ESD-TR-77-162

Semiannual Technical Summary

Network Speech Processing Program

31 March 1977

Prepared for the Defense Communications Agency
under Electronic Systems Division Contract F19628-76-C-0002 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LEXINGTON, MASSACHUSETTS

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, for the Military Satellite Office of the Defense Communications Agency under Air Force Contract F19628-76-C-0002.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Raymond L. Loiselle

Raymond L. Loiselle, Lt. Col., USAF
Chief, ESD Lincoln Laboratory Project Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

NETWORK SPEECH PROCESSING PROGRAM

SEMIANNUAL TECHNICAL SUMMARY REPORT TO THE DEFENSE COMMUNICATIONS AGENCY

1 OCTOBER 1976 – 31 MARCH 1977

ISSUED 25 AUGUST 1977

LEXINGTON MASSACHUSETTS

ABSTRACT

This is the first Semiannual Technical Summary Report on the Network Speech Processing Program to be submitted to the Defense Communications Agency. It covers the period 1 October 1976 through 31 March 1977 and reports on the following topics: Secure Voice Conferencing, Speech Algorithms, and Bandwidth Efficient Communications. Each of these tasks is directed to particular problems associated with AUTOSEVOCOM II and/or the design of future defense communications systems.

CONTENTS

	Abstract	iii
I.	INTRODUCTION AND SUMMARY	1
II.	SECURE VOICE CONFERENCING	3
	A. Introduction	3
	B. Interim Conference Capability	3
	1. Hardware Facility	3
	2. Conference Control Algorithms	4
	C. Large Conferencing System – The Secure Voice Conferencing Facility	5
	1. Hardware Facility	5
	2. Software and Experiments	6
	D. Human Factors Methodology	7
	E. Preliminary Conferencing Experiment Results	9
	F. Future Experiments and Demonstrations	9
	1. Future Experiments	9
	2. Demonstrations	10
III.	SPEECH ALGORITHMS	11
	A. Introduction	11
	B. LPC AGC Algorithm	11
	C. LPC-Belgard Experiment	12
IV.	BANDWIDTH EFFICIENT COMMUNICATIONS	15
	A. Introduction	15
	B. Model of a Single Link in a PVC Network	15
	1. PVC Network Features	15
	2. Model Description	20
	C. Behavior of Voice and Data Traffic in the PVC Link Simulation	21
	1. Voice Queue	21
	2. Data Queue	21
	D. Additional Measurements on a Single PVC Link	24
	1. Varying Data and Voice Priorities	24
	2. Techniques of Discarding Packets and the Size of the Voice Queue	25
	3. Behavior of Smaller-Capacity Links	25
APPENDIX A – The Secure Voice Conferencing Facility		33
APPENDIX B – A Resource Allocation Task		43

NETWORK SPEECH PROCESSING

1. INTRODUCTION AND SUMMARY

The Network Speech Processing effort funded by DCA consists of three major tasks focusing on secure voice conferencing, narrowband speech digitizing algorithms, and bandwidth efficient communications. Each of these tasks is directed to particular problems associated with AUTOSEVOCOM II and/or the design of future DCS.

The secure voice conferencing effort is concerned with the analysis and simulation of various conference bridging and switching configurations, control protocols, delays, and conference sizes.

The effort in speech algorithms is presently directed toward the severe problem of wide speaker dynamic range which causes distortion in narrowband speech digitizers (specifically LPC vocoders). Solutions to this problem will also enhance the quality of speech realized from LPC-CVSD and CVSD-LPC tandem connections.

The ongoing study of bandwidth efficient systems, in particular the packetized virtual circuit (PVC) approach, is yielding valuable data on circuit and system utilization and efficiency, network delays, and various sources of distortion.

The following sections describe progress to date on each of the three contract areas. The conferencing section discusses the large conference facility designed to simulate all the conference configurations pertinent to DCS, as well as outlining the human factors evaluation issues. The speech algorithm section discusses present results in automatic-gain-control experiments applied to the LPC vocoder. Finally, the bandwidth efficient communications section presents results to date on all the PVC studies performed by Lincoln Laboratory.

II. SECURE VOICE CONFERENCING

A. INTRODUCTION

The objectives of this part of the DCA effort are the analysis of various methods of secure voice conferencing, appropriate demonstrations of the most promising of these methods, and, finally, recommendations for conference techniques to be incorporated in AUTOSEVOCOM II and the Worldwide Secure Voice Architecture. Our interim experiments, described in detail later, have included several broadcast techniques for digitized teleconferencing, issues of push-to-talk (PTT) vs voice energy switching, and the effect of transmission delay on teleconference quality.

The thrust of the Lincoln Laboratory effort in FY 77 has been toward the completion of a large-scale conferencing test bed – referred to as the secure voice conferencing facility – which has been designed for simulation of all the desired teleconferencing configurations of interest to the DCA. The new system allows for up to twenty conferees connected to a combined signal processing switchboard and PDP 11/45, as well as a control input channel. The signal processor will implement all the audio, delta modulation, or frame combining and switching conference bridges. External voice digitizer equipments will add tandemed conferencing capability to the simulation system.

An interim conferencing system originally designed for four users in a PTT control environment has been modified to allow full-duplex and modified broadcast interaction between conferees. Delay can be introduced as a separate controlled parameter. This interim system permits us to consider some questions about limited numbers of teleconferees using controlled and full-duplex, delayed, and undelayed systems. In addition, the system has supported human factors studies of teleconferencing situations. The human factors studies have been undertaken under subcontract to Bolt Beranek & Newman, Inc. of Cambridge, Massachusetts, working closely with Lincoln staff and using the Lincoln Laboratory systems. When the large conferencing system is operating, these studies will shift to the examination of issues important to large teleconferences, using the methodologies acquired in this interim period.

In Sec. B, we discuss the interim conference test bed hardware, its modest capabilities, and some of the experiments we have designed. In Sec. C, we present an overview of the secure voice conferencing facility hardware and discuss our initial series of teleconferencing simulations. The BBN approach to the conferencing evaluation problem is outlined in Sec. D. Preliminary results and future experiments are discussed in Secs. E and F, respectively.

B. INTERIM CONFERENCE CAPABILITY

1. Hardware Facility

A hardwired conference configuration was designed for an earlier study of ARPANET teleconferencing. This configuration is composed of four teleconference stations, each in a different office. Each station consists of a telephone handset and a small control box containing two pushbuttons for signaling from station to computer, and two lights for signaling from computer to station. Each of these four stations is connected to a central signal conditioner and interface box through a 9-conductor cable (audio transmit pair, audio receive pair, 2 lights, 2 buttons, ground). The interface box communicates with a PDP 11/45 processor through a standard DR11C interface card, enabling two-way interaction between conference configuration and computer.

The original function of the central signal conditioner and interface box was that of switching one of the four talkers as the input to a narrowband speech digitizer. The output of the digitizer was then broadcast back to the other three conferees. This switching was accomplished with a set of analog gates, controlled by the computer, so that handset microphones and receivers were connected to the input and output of the speech digitizer as defined by the conference protocol. As a conference parameter, delay could be introduced at the digitizer level. In addition to the audio amplifiers and gates for the transmitter and receiver connections, the interface box contained sufficient logic to signal the computer when a conference station button was pushed, to request or relinquish connection to the digitizer (start or stop talking). The computer could also light or flash one or both of the conference station lights by outputting an appropriate bit.

This hardware configuration continues to be used as our interim conference facility. In its old form, it can be used to explore PTT conferences of four users or less. The actual narrowband speech digitizer used to support the conference can be LPC, CVSD, APC, or clear audio depending on the requirement. Modifications to the configuration allow us to tap off each of the audio transmit and receive pairs and drive separate speech digitizer equipments whose digital outputs can be combined, as described in the next section. This allows us to explore other broadcast conference techniques in addition to PTT, and to compare the behavior of PTT and modified full duplex under various delay conditions.

2. Conference Control Algorithms

All our experiments to date have used one or another form of signal switching as opposed to mixing, or bridging. We focused our attention on switching techniques during this interim period because the available hardware was better suited to switching experiments. We have explored two basic types of switching algorithms: PTT, and voice-controlled switching (VCS). In the following sections we describe the algorithms available for experimentation.

a. Push-to-Talk Algorithm (PTT1)

Our PTT experiments make use of the hardware described in Sec. B. 1 above. When a participant wishes to speak to the conference, he pushes the green button on his control box. If no one else is speaking, he will be given the "floor," so to speak. This situation will be indicated to him by the lighting of the green light on his control box. While he has the floor, he also hears his own voice as a side tone in his receiver. If someone else is speaking when he pushes his green button, his request to speak will be put in a queue by the control program and the red light on his control box will flash to signal the successful receipt of his request by the control program.

Once a participant has been given the floor and is speaking to the conference, he may continue to do so until either he finishes with what he has to say and so indicates by pushing his red button, or the control program decides that he has talked long enough and gives the floor to another participant who has been waiting to talk. In the latter event, the green light on his control box will flash for 30 sec as a warning before the floor is taken away from him. The warning light will not start flashing until the speaker has talked for 30 sec. As a result, each participant is guaranteed at least 1 min. without interruption. Once he has lost the floor, he must push his green button again to get back in line for another chance to talk.

The red button can also be used to cancel a request to speak while waiting in the queue. Such a cancellation is not an uncommon event in many conference situations.

It should be noted that our PTT situation is different from that in many communication systems in that the participant does not have to hold the button depressed while he is talking.

There is no priority structure in effect in our present PTT conferencing algorithm (PTT1). Participants are given the floor in a first-come first-served fashion. While the software allows for pre-emption by one of the participants acting as a chairman, our experiments to date have not made use of this feature.

b. Voice-Controlled Switching Algorithms (VCS1, VCS2, VCS3)

We have established a facility for examining frame synchronously, the analysis from four independent LPC vocoders (2400 bits/sec) and assigning coefficients for the synthesizers based upon the LPC energy measure of speech activity. An independent decision is made for each listener every 20 msec vocoder frame time. In the case where only one speaker has an energy measure above a speech activity threshold, his speech will be broadcast to the other conferees. If more than one conferee has energy above threshold, a conflict is said to exist and the control program must decide which speaker each listener is to hear. (A speaker never hears his own vocoded speech.) In a two-speaker conflict, the coincidentally active speakers hear each other as though they had a full-duplex connection, but the quiet listeners hear some combination of the two speakers' speech. We have explored three possibilities for deciding what the quiet listeners should hear, namely:

VCS1	The listener hears one frame at a time from each of the conflicting speakers for the duration of the conflict.
VCS2	The listener always hears the frame from the speaker with the greatest energy.
VCS3	When a switch of speakers occurs, the listener will hear the speaker with the greatest energy, but further switching will be inhibited for 1/2 sec unless the new speaker's energy falls below the speech activity threshold.

With VCS1 and VCS2, the listener can easily discern that someone is attempting to interrupt the speaker, but the intelligibility of the speaker is seriously reduced during the conflict interval and neither the identity of the interruptor nor the content of his speech is likely to be recognized. With VCS3, however, since the switching interval allows units the size of syllables (and sometimes whole words) to pass through undamaged, the interruptor is likely to be recognized and in some cases the content of his speech will also be understood. Since VCS3 is by far the best of the voice-controlled switching algorithms we have explored to date, we have employed this technique in all our human factors experiments.

As an additional parameter, we can simulate uniform satellite transmission delays on the order of seconds at present, with differential delays among participants available in the near future.

C. LARGE CONFERENCING SYSTEM – THE SECURE VOICE CONFERENCING FACILITY

1. Hardware Facility

When the requirement for 20 audio connections was examined in the context of the various teleconference configurations spelled out in the DCA statement of work for FY 77, it became

clear that a dial-up phone arrangement was preferable to a hardwired facility. The nature of some of the conference bridging requirements also led us to a more general approach than that used in our earlier system. The final conferencing facility designed to fulfill all the requirements of our FY 77 work statement is described in detail in Appendix A.

The secure voice conferencing facility consists of two separate subsystems connected to the PDP 11/45 computer (as shown in Fig. A-1 of Appendix A). A touch-tone signaling and modem system is connected to ordinary telephone lines and provides dial-up answering and tone decoding for establishing audio connections and decoding control signals from users. A second subsystem provides an elaborate switchboard-signal conditioning capability by providing A/D-D/A channels feeding and being fed by a signal processing computer (the Lincoln Laboratory DVT). This subsystem will implement most of the actual conference bridging techniques using the digital speech samples, and then provide the bridge output to various users through D/A converters. A large-core memory allows the signal conditioning to introduce delays of up to 1/2 sec for each of 20 audio lines, approximating a two-hop satellite communications delay. Hybrid "two wire to four wire" transformers split the incoming audio phone pair into separate input and output pairs to connect to A/D and D/A converters. Examples of conference system applications are presented in Appendix A.

2. Software and Experiments

The first experiment we anticipate running on the secure voice conferencing facility will be the baseline system of a single audio summing node which is distributed to each listener. In order to minimize listener-talker echo effects (especially under delay conditions), each listener will hear the summing node minus his own input. The system is a true full-duplex system since any two talkers can communicate through the summing node without hearing his own signal. This system is sometimes referred to as a free-for-all system since there are no constraints or priorities on talkers. In order to implement this system we not only need the hardware discussed in Sec. 1 above, but software running in two machines as well. In the PDP 11/45 we provide control and decode functions for the touch-tone receiver-modems. This software also provides the initial answer response to dial-up signals. Then the touch-tone symbols must be decoded by table lookup, and the information transmitted to the software handling the audio conditioner computer (the DVT). For example, a user will dial up the facility, be answered, then touch-tone request to join a full-duplex conference. His request will be signaled to the DVT by way of PDP 11/45 software, and the DVT will start accepting data samples from the dialed-up user's A/D channel. The software running in the audio conditioner signal processing machine (the DVT) will actually perform the conference bridging. In each 5-μsec sampling interval (see Fig. A-3 of Appendix A), an activity flag is tested, a new A/D sample is received, a delay may or may not be introduced, a summing node is outputted to the D/A (minus the input from this channel), the summing node is updated, overhead indexing takes place, and the computer is set for the next 5-μsec sampling interval (next talker). After several of these sampling cycles have taken place (i.e., several of the 125-μsec intervals), the DVT will transfer as many as 20 words to the 11/45 and receive an updated set of as many as 20 status words. This interaction occurs during slot 24 and 25 dead time (see Fig. A-3 of Appendix A) and allows the DVT to provide real-time statistical data (e.g., talk spurt durations) to the 11/45, and the 11/45 to update talker activity on the audio conditioner switchboard.

The second experiment to run on the facility will be a single stream audio broadcast system. This arrangement of the conferencing facility will permit PTT control (eventually energy-detector control) over who is broadcasting. To accommodate this form of conferencing, the PDP 11/45 software will be augmented to respond to requests to speak, and to requests to stop speaking. This response will take place through the touch-tone modem interface software translator, and will then be passed on to the software delivering status words to the DVT. Flag bits in these status words will indicate which channel is to be considered the broadcaster, at the DVT level. Additionally, the touch-tone handler will control tone response to users to indicate that the conference is listening to them, as well as tone response to indicate that the conference has finished listening to them. At the audio conditioner level the single talker can be broadcast directly to all listeners, or processed through voice digitizer hardware such as LPC, CVSD, or APC before broadcast back to the other conferees. A second stream accommodating a second talker (really an interrupter) can be set up so that, upon touch-tone request and response, the talker who is broadcasting will hear the second talker and respond to him, or relinquish the floor so that he may become the broadcaster.

Figure A-6 in Appendix A indicates the flow of voice for a CVSD bridging which is taking place in the DVT software. Again, the control path to establish such a conference will take place by way of the dial-up network connected to touch-tone modems and interface. The PDP 11/45 touch-tone handler will decode and establish active channels for the DVT running the bit-combining CVSD code. Figure A-6 also indicates the use of external vocoder hardware to simulate a tandem quality conference.

With the large conferencing facility flexibility, it will be possible to continue to implement the experiments discussed in Sec. E below until we have reached suitable design criteria and demonstrations of feasibility for defense communications systems.

D. HUMAN FACTORS METHODOLOGY

Although real-world conferences among parties remote from each other have been conducted for years, scientific study of such "interaction-at-a-distance" has only recently begun. As a result, much of the data required for evaluation of primary and interactive effects of such variables as conference size (number of conferees), conference task (information collection, dissemination, problem solving), control discipline (chairman vs agreed-on "polling" procedures vs no formal discipline), etc. on system design are not yet available. Moreover, empirical and theoretical findings reported in the otherwise relevant literatures, of human factors, artificial intelligence, social psychology, management, speech, and hearing, do not exist in a form in which they can be exploited by architects and users of teleconferencing systems.

The small amount of data in the area, together with BBN experiences in the conduct of research in information processing and decision making, suggest that a key requirement for successful evaluation of teleconferencing alternatives is the availability of laboratory tasks that meet the following criteria:

(1) A given problem scenario should be usable over the entire range of conference sizes to be evaluated, and its difficulty level should be independent of size. Furthermore, scenarios should be constructed in such a way that they can be reused with a given set of conference participants.

(2) Problems selected should be intrinsically interesting to subjects, and the testing situation should promote highly motivated performance.

(3) Scenarios employed should permit a variety of objective performance measures, including gross measures such as solution time and solution quality, and fine measures of communication and system effectiveness and dynamics, such as number of messages per speaker per unit time, average queue length and speaker waiting time, and duration of pauses between messages.

The BBN effort during Phase I of the project has been devoted almost exclusively to the development and preliminary test of laboratory tasks that satisfy the above criteria. This has resulted in the tentative selection of four different tasks that, we expect, will make up a portion of a larger battery to be administered during the formal evaluation to be conducted in Phase II. In addition to meeting the formal criteria, each has so far proved to be an efficient generator of data – that is to say, each promotes a high rate of information exchange and encourages approximately equal contributions by conference members. Additional characteristics of interest at this point are (1) that all tasks require the cooperative interaction of conference participants in order to reach effective solutions, and (2) that at least two of the four tasks appear to provide vehicles in which teleconferencing objectives such as information collection, information dissemination, and dynamic exchange of information can be addressed analytically. A brief description of one of these tasks, in which conferees attempt to achieve an optimal allocation of resources in accord with specified constraints, is provided in Appendix B of this report.

With the exception of one trial run on the resource allocation task, all tests to date have been conducted with the same set of subjects. With one more exception, all conferences have been limited to three persons. Finally, only three system conditions – PTT1 with a delay of 500 msec, and VCS3 with delays to all speakers of 80 and 500 msec – have been reviewed. This focus has provided considerable flexibility in the design and redesign of problem scenarios, and has enabled possible rejection of one entire category of tasks earlier thought to be desirable as experimental vehicles – viz., those which produce a high rate of speaker interruption by requiring conference members to compete with each other for information.

With the beginnings of a test battery in hand, it is now becoming critical that we consider questions related to the formal administration of conferencing experiments and to the analysis of data proceeding therefrom. This consideration, which will proceed concurrently with the further development of the test battery over the next month, will be concerned essentially with the accommodation of two conflicting constraints: (1) the need to generate valid and reliable data for a potentially large number of system, delay, conference size, control algorithm combinations; (2) the necessity for piecewise development of system capabilities. What we must evolve is a design which either minimizes effects due to the order in which subjects are confronted with experimental treatments, or which ensures that order effects, if any, can be treated statistically during analysis. We recognize that specification of a design which will achieve the required control over sequence effects can be eased, at least in principle, by the inclusion of formal training and practice sessions prior to exposure to each new experimental condition, and by periodic comparison of performance within a given condition with that obtained concurrently in the baseline (analog bridge) condition. However, it seems clear at this point that recourse to either of those techniques will have a significant impact on the length of time required to complete the evaluation.

E. PRELIMINARY CONFERENCING EXPERIMENT RESULTS

Although no opportunity for formal experimentation was afforded during the first month of the BBN subcontract, testing of problem scenarios and the familiarity gained as a result of that testing have led to a number of tentative observations regarding conferencing systems. The first is that, by modifying their behavior, conferees will find a way to cope with system limitations confronting them. If the problem to be solved is made more difficult by planned or inadvertent interruption of speakers, conferees will voluntarily slow down their rate of interaction, establish a polling procedure, respond to a de facto leader, or develop some other strategy that appears to promote the business of the conference. If, for one reason or another, a conferee cannot be clearly heard by others, he is informed of the fact and he then speaks more loudly or more slowly. Such adaptability is obviously to be expected, and it could have an important impact on the extent to which the systems can be differentiated on the basis of objective measures of efficiency. If, for example, conferees are able to compensate effectively for the need to change their rates of speech by manipulating the information content of their utterances per unit time, data with respect to frequency of interaction, problem solving performance, queue length, etc. may be difficult to interpret. In such circumstances, subjective evaluations of ease of use may provide as appropriate a yardstick for relative system desirability as can be found.

Second, we have observed some asymmetry in the relative ease with which the PTT1 and VCS3 systems can be used. The latter system requires little or no consideration on the part of the conferee with respect to requirements for establishing or maintaining a dialog. His attention can be allocated almost exclusively to the business of the conference, and he contributes, more or less naturally, as required; PTT1, on the other hand, appears to require a fairly significant division of attention. Unless a polling procedure has been established, the conferee must allocate attention between the visual display of lights and the materials (maps, score sheets, etc.) relating to his intended input. Moreover, he must give some consideration to the timing of his intended input in the event that he cannot secure the channel directly. The full impact of this division of attention is not clear to us at the moment, but we believe it may be a detriment to conference environments in which freely flowing discussions (as opposed to simple exchanges of information) are desired.

The final observation to be made here is that there appears to be no difference in performance as a result of delays up to 0.5 sec. However, since all speakers are delayed by the same amount under current conditions, it is impossible to assess the impact of a single disadvantaged participant on total conference performance. We believe that this may be an area where the earlier discussed adaptability of conferees to adverse conditions may successfully mask the negative effects of all but the longest possible delays.

F. FUTURE EXPERIMENTS AND DEMONSTRATIONS

1. Future Experiments

With the A/D-D/A audio conditioner subsystem connected to the conference structure and completion of the touch-tone controller software about the same time (written for the PDP 11/45 in the Unix operating system), we anticipate that the system will be fully operational to run a baseline, audio summing full-duplex conference, running in the DVT. A single-stream audio broadcast conference experiment with touch-tone PTT will then be simulated, followed by a dual-stream broadcast experiment. The dual stream allows the primary broadcast talker to

hear an interrupter on a second stream while the other conferees continue to hear the primary talker. With completion of the large memory interface hardware along with diagnostics, we may begin adding delay effects to our active conference experiments. Further work in voice energy switching of clear audio, CVSD energy switching, and CVSD bit combining-bridging will follow. Finally, we will be able to simulate a large LPC conference with frame combining by using voice switching of 10 to 20 conferees into 3 or 4 LPC vocoders.

2. Demonstrations

Because of the dial-up and touch-tone control capability of the secure voice conferencing facility, we will be in a position to arrange several demonstrations which can be dialed up and used from DCEC Reston or any other convenient location.

Two demonstrations are planned for late FY 77. A full-duplex audio bridge conference in mid-August, with or without adjustable delay, can be demonstrated in dial-up mode from Reston. This demonstration serves to display the basic capability of the conferencing facility. This will be followed by a PTT broadcast type conference which will be ready for use from Reston in mid-September. Again, this configuration will also allow for demonstration of delay effects. Later in the calender year, a demonstration of CVSD bit combined or energy switched conferencing will also be available.

III. SPEECH ALGORITHMS

A. INTRODUCTION

The Consortium tests have demonstrated that, in a relatively benign environment, LPC vocoders offer satisfactory intelligibility and quality ratings. However, in a more severe environment, the suitability of LPC is still open to question. In particular, when LPC is placed in tandem with the AUTOSEVOCOM II standard (CVSD), appreciable degradation results. Thus far, attempts to increase acceptability of the LPC, CVSD tandem have been directed toward the following two approaches: one is to try to improve the match between the LPC output and the CVSD input (and vice versa), and the other is to try to improve the intelligibility and quality of the individual devices. Previously, we had developed the idea of chirp filtering to enhance the LPC output – CVSD input match; Secs. B and C below describe two methods for individual improvement.

B. LPC AGC ALGORITHM

One major complaint about LPC is the distortion produced with abnormally loud or soft speakers. The cure for this seems to be some form of AGC to increase the input dynamic range over which LPC can produce good-quality speech. The distortion caused by loud speakers can be cured by adjusting the input gain to the A/D converter so that the loudest expected speaker does not cause A/D clipping. Normal and soft speakers will now only use a fraction of the available A/D dynamic range, and thus cause a possible degradation in LPC quality. This loss of quality can be due to two causes: increased input quantization noise, and loss of significance in the LPC analysis calculations. The quantization noise problem can be cured only by using an A/D converter with a larger word size or using a program-controlled attenuator at the input to the usual 12-bit A/D converter. One important result of the present AGC investigation is that quantization noise does <u>not</u> seem to be a problem, thus obviating the need for these hardware cures.

The second problem, loss of significance in the analysis calculations, can be solved by suitably upscaling the speech before analysis. Figure III-1 shows the procedure chosen to accomplish this end. The scale factor for a given frame is determined by first finding the maximum speech sample in the frame. The number of bits (up to a maximum of 4 bits) that this maximum sample can be left shifted without overflow is the scale factor for that frame. The incoming speech is delayed by one frame so that the scale factor just determined can be applied to each sample in that frame. It is important to note that this form of scaling in no way distorts the input speech waveform within a frame. The frame's scale factor is also used to dynamically vary the buzz/hiss threshold in the pitch detector in order to prevent low-level input signals from forcing the pitch detector to erroneously declare hiss. An attempt was made to use the upscaled speech in the pitch detector, but this produced unpleasant artifacts in the output speech.

The scaled speech is now analyzed with the conventional LPC algorithm, and the parameters are coded and shipped to the receiver. An additional 2 bits describing the scale factor are sent along with the other parameters. After the decoding operation at the receiver, the residual energy parameter is subject to a downscaling based on the scale factor. The remainder of the processing is the normal LPC synthesis algorithm.

The downscaling operation has been the subject of intensive investigation. There are two issues here: where in the analysis-synthesis chain to perform the downscaling, and what strategy to use for downscaling. Three places for downscaling suggest themselves – before coding, after decoding as shown in Fig. III-1, and after synthesis. All three ideas were tried, and downscaling after decoding was judged to be the best method. Downscaling before coding worked well with high-level input signals, but performed poorly with low-level inputs because the residual energy was too low in the coding table. Downscaling after synthesis was rejected because it yielded a "bumpy" sounding output because of frame-to-frame discontinuities in the synthetic speech.

The first downscaling strategy that was tried was a linear one, i.e., the excitation amplitude was downscaled by the same number of bits that upscaled the input speech in the current frame. This algorithm produces speech that is indistinguishable from ordinary LPC speech until the input level drops to the point where ordinary LPC drops into steady hiss. Both algorithms are still perfectly intelligible at this point. This state prevails until the input level becomes so low that the residual energy for the ordinary LPC algorithm becomes too small for the coding table. The AGC version of the algorithm is still highly intelligible, but the output level is too low to be useful.

The next downscaling strategy that was tried was a nonlinear algorithm, i.e., the excitation amplitude was downscaled by an amount dependent on, but not equal to, the amount the input speech was upscaled. The particular algorithm chosen compressed amplitude levels at low and high levels, but was linear for intermediate levels. Initial tests of this strategy have been most encouraging. At input levels low enough to cause ordinary LPC to suffer severe energy quantization effects, the AGC version of the algorithm still produces excellent quality speech at an acceptable output level. Further work is being performed in this area.

C. LPC-BELGARD EXPERIMENT

A new type of vocoder algorithm has been proposed to try to overcome the limitation that a conventional LPC vocoder cannot model spectral zeros. A block diagram of this algorithm appears in Fig. III-2. The basic idea is to use LPC techniques to generate a residual error signal which is then analyzed by a channel vocoder filter bank. The parameters shipped to the receiver are the LPC reflection coefficients, the channel vocoder parameters characterizing the spectral envelope of the error signal, and pitch derived from a pitch detector working directly on the input speech. At the receiver, the pitch word is used to generate the excitation for the channel vocoder synthesizer whose output should be an approximation to the error signal. This synthetic error signal is now used to excite a conventional LPC synthesis filter whose output is then the final synthetic speech.

The implementation of this algorithm on a DVT is a nontrivial task because the Belgard algorithm alone uses most of the DVT's resources, both with respect to running time and memory occupancy. This means that it is impossible to run the algorithm in a single DVT. Using two DVTs, one for LPC analysis/synthesis and another for Belgard analysis/synthesis, is also not feasible because it requires the DVT running the LPC algorithm to input two samples and output two samples every 132 μsec. The DVT's limited I/O capability makes this impossible to do. These considerations led to the experimental setup shown in Fig. III-3. DVT1 will accept the input speech and use LPC analysis to produce the residual error signal. The latter will be shipped to DVT2 via DVT1's D/A port where it will be subjected to Belgard analysis/synthesis.

Belgard's pitch detector will be used to derive the pitch from the error signal rather than from the raw speech, as was discussed earlier. The resulting synthetic error signal will be shipped to DVT3 via DVT2's D/A port. In addition, DVT1 will serialize the reflection coefficient parameters and ship them to DVT3 via the parallel/serial-serial/parallel (P/S-S/P) interface. These parameters will then be unpacked and used in conjunction with the incoming synthetic error signal from DVT2 to produce the final synthetic speech.

This arrangement was implemented, and it was determined that use of the serial channel to ship the reflection coefficients to DVT3 requires buffering, rate control, and synchronization protocols which introduce time-varying relative delays between the reflection coefficients and the synthetic error signal. These delays produce unacceptable artifacts in the output speech. These artifacts are present even when the raw error signal produced by DVT1 is shipped directly to DVT3 without Belgard intervening.

Since this work was performed, the LDSP has been completed, and its A/D-D/A converter is now nearing completion. When this occurs, the LDSP will have the capability of accepting two analog input streams and delivering two analog output streams. With this capability, the algorithm can be tested with the setup shown in Fig. III-4 which is algorithmically the same as that of Fig. III-3 except that now LPC analysis and synthesis will be done in one machine, the LDSP, thus eliminating the need for the serial data path and its attendant delay problems. We expect that this arrangement will be tried in the near future.

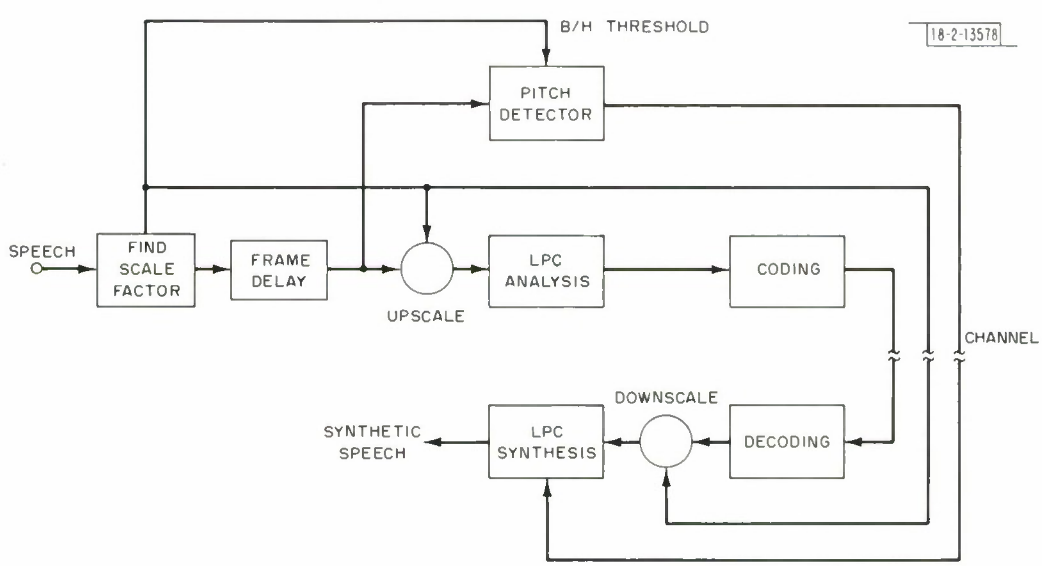

Fig. III-1. LPC AGC system.

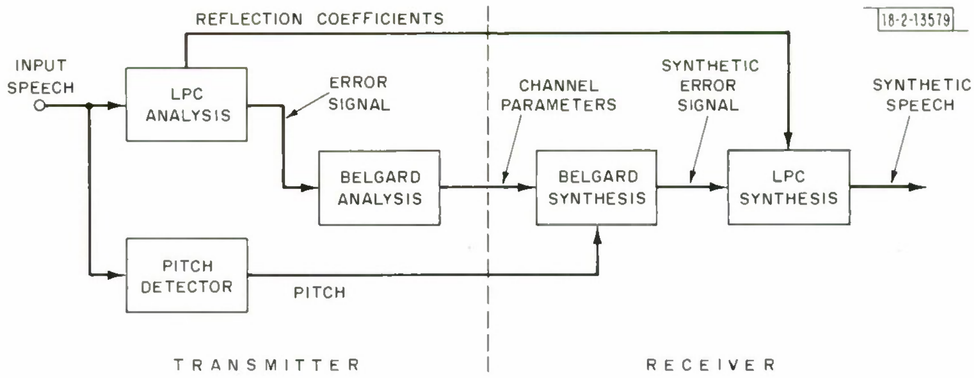

Fig. III-2. Proposed vocoder algorithm.

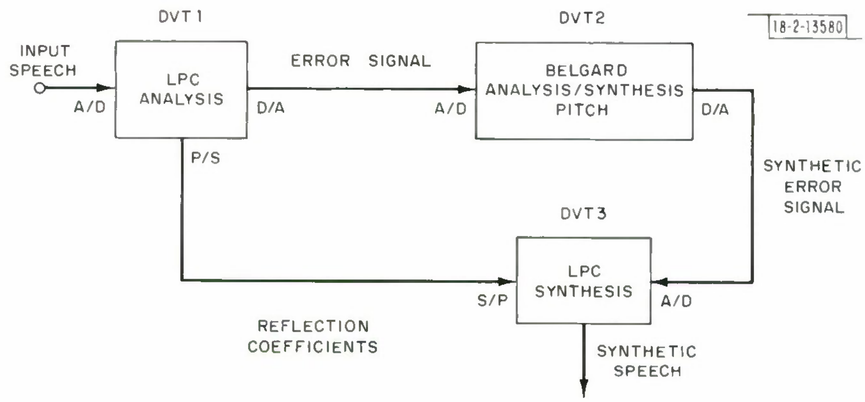

Fig. III-3. Vocoder implementation.

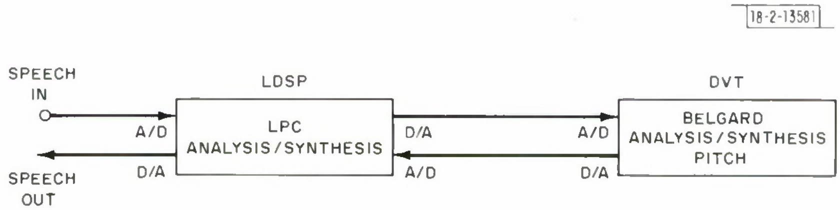

Fig. III-4. Proposed vocoder implementation.

IV. BANDWIDTH EFFICIENT COMMUNICATIONS

A. INTRODUCTION

Packetized Virtual Circuit (PVC) techniques combine features of both circuit- and packet-switching technologies to provide a very efficient approach to integrating voice and data in a communications network. The motivation for integrating voice and data in a communications system lies both in expected cost savings derived from the sharing of transmission and switching facilities, and in the promise of greater flexibility in coping with changing traffic patterns. While most of the proposals for integrated networks have involved some mixture of circuit-switching techniques for voice and packet switching for data, the PVC approach[1] handles both types of traffic in an essentially uniform fashion, easing the implementation and providing the capability to respond automatically to changes in traffic mix. The PVC network concept attempts to capitalize on the statistical multiplexing advantages inherent in packet technology. At the same time, it attempts to overcome some of the efficiency and delay dispersion difficulties associated with pure packet networks by utilizing communication link formats and routing conventions associated with digital circuit switching. Since the flow-control mechanisms normally employed in packet networks introduce delays and loss of efficiency which are inappropriate in a network intended to handle a high percentage of voice traffic, the PVC network depends on a flow-control discipline that has been termed "statistical flow control." In addition, the PVC system is designed to take advantage of the on-off statistics of voice traffic to increase its capacity to handle both voice and data.

The PVC approach requires the establishment of a connection from source to destination hosts, fixing most of the packet header information. All packets in the connection follow the same path through the network. The PVC packet header need only contain information identifying it as belonging to a particular connection. Thus, packet overhead is reduced significantly by the use of connections, and short packets can be employed efficiently.

In the PVC scheme, flow control is performed by the assignment of connections to specific links to reduce the probability of internal overloads to values that are small. This permits treating the problems caused by the overloads on an exceptional basis without introducing severe overhead. This new and untried approach to flow control is a vital factor of the PVC network concept.

Packet-speech techniques[2,3] provide a straightforward way to take advantage of silent intervals, which represent more than half of the elapsed time in typical conversational voice transmissions. By using a speech activity detector and not sending packets when no activity is detected, a packet voice network could expect to handle roughly twice the voice traffic that could be handled if the nominal voice encoding rate had to be handled continuously – that is, by a circuit switched network with the same channel capacity.

B. MODEL OF A SINGLE LINK IN A PVC NETWORK

1. PVC Network Features

The basic features of a PVC voice/data network are described in this section. The single PVC link that was simulated is based upon these features.

a. The PVC Packet

The PVC net is designed to use a small fixed-size packet for all communication. The use of a small packet tends to reduce both average delays and the dispersion of delays, a feature required for good voice transmission. Unfortunately, the use of small packets also tends to reduce communication link efficiency (each packet must have its own header), and special measures must be taken to keep it at workable levels. The fixed packet size allows the use of a frame structure on the communication links. By counting bits from a frame marker to locate the beginning and ending of packets, the PVC net minimizes the need for channel capacity to communicate packet-location information. The channel capacity required for frame markers can be kept quite small and has been assumed to be negligible in the simulations. Since there is no use of the frame structure other than to identify potential packet slots, it is not included in the model.

A packet size of 128 bits has been chosen more or less arbitrarily. The exact value is not critical, and there appears to be no mathematically determinable optimum value in terms of any simple measures of network performance. Of the 128 bits, it is postulated that 32 would be used to carry the necessary forwarding information, sequence number, and check bits. The resulting packet efficiency would be 75 percent. In the case of voice traffic, it might be possible to reduce the header and trailer bits in the packet to 16 if bit errors could be tolerated in the voice encoding scheme in use. The efficiency for voice packets would then be 87.5 percent, but since the desirability of allowing such errors is questionable, the more conservative 75-percent figure has been used in the calculations.

In order to reduce the packet overhead to as small a value as 32 bits, the network utilizes the virtual connection convention with all packets belonging to a connection following the same route through the net. In this case, the other information needed to completely identify the packets (type, priority, source, and destination addresses) can be communicated during the process of setting up the virtual connection, and an abbreviated header can be used to identify the packets as belonging to a particular virtual connection.[4] By the use of tables at each forwarding node, the field used for identifying the packet, called the forwarding address, may be further reduced in length to a value just large enough to include the maximum number of connections which can be routed on any given link. This number is obviously dependent upon the link channel capacity and upon the intended use of the network. For example, if one assumes a 1.544-Mbps link such as has been considered in the simulations, and if 16 kbps were the lowest data rate in use for voice encoding, 8 bits would suffice for the forwarding address in all voice connections which could be handled by the link. Lower expected data rates for data traffic would require a larger field for the forwarding address of such packets. In the extreme case of a very large number of low data rate terminals, the required forwarding tables would become burdensome, and a concentration of terminal streams into virtual connections would be indicated. It has been assumed that a forwarding address field of 12 bits would suffice for reasonable mixes of data connection on a 1.544-Mbps link.

It should be noted that, while the basic packet efficiency of a PVC net under the above conditions would be 75 percent, the actual efficiency would be somewhat less due to the fact that packets would not always be full. Further, the overhead of end-to-end acknowledgments and possible retransmission of data packets should be taken into account. The present studies have not yet progressed to a point where estimates for the magnitude of these overhead quantities can be

stated with any confidence. However, since hop-by-hop and end-to-end acknowledgments are not automatically generated on a packet basis as is customary in conventional packet networks, we expect that the overall efficiency of the PVC net will be significantly superior to that observed in pure packet networks.[5]

b. Packet Forwarding

Because the PVC packet is short, the time available for forwarding processing is also short (83 µsec for 128-bit packets on 1.544-Mbps links). The use of a forwarding table unique to each link coming into a node is indicated. The forwarding table would be indexed on the incoming forwarding address and would contain the following items:

(1) The output link to be used by the packet. If the node were the destination node for the connection, this item would indicate the port to be used by the exiting packet.

(2) The new forwarding address to be used on the output link or port. The forwarding address must, in general, change from link to link because the address field is not large enough to encompass all active connections in the net.

(3) Connection type (and priority if required). This information is used in deciding into which output queue a packet is to be placed. It is also used in the process of taking down the connection.

(4) Average data rate required by the connection. This information is used to inform the routing process as to the link capacity made available when the connection is taken down.

(5) A back pointer to facilitate the operation of taking down a connection. This pointer is necessary to permit tracing the connection back from the destination to the source. (In the PVC net, a virtual connection is a one-way path from source to destination.)

Altogether, forwarding tables might require as many as 48 bits per entry and have 4000 or so entries per link in a network designed to handle many simultaneous data connections. The cost of memory for such tables does not appear to be prohibitive in relation to the high performance of the network. Further, the processing power needed to effect the table lookup and carry out the forwarding process is very modest, and software or firmware requirements are modest.

It is postulated that traffic in a PVC net could be adequately handled by defining three classes of packets – voice, data, and supervisory – which would be held in separate output queues. Supervisory packets would be provided with a small fixed fraction of the channel. In the simulation, a value of 8 kbps was chosen. Voice packets would be given priority in the forwarding process because of their real-time requirements. Finally, data packets would be sent in the remaining packet slots.

c. Routing

Routing problems in the PVC net are very similar to those encountered in conventional circuit-switched networks. When a subscriber requests a virtual connection, a path must be

found between source and destination ports which has sufficient uncommitted capacity to handle the requested data rate. If such a path cannot be found, the request will be rejected, in which event the subscriber has the option of requesting a connection at a lower data rate or waiting until the network is less busy.

Since the problem of finding a path in a circuit-switched network subject to various constraints and cost criteria has been investigated at length by others, and the PVC net seems to pose no unique problems in this regard, we have not yet focused our attention on the selection of any particular routing algorithms for use in the PVC net. The principal difference between PVC and a circuit-switched net in the routing area lies in the use of average rather than peak data rate requirements in determining whether or not sufficient capacity remains in a link being considered as a possible route for a connection. By scheduling traffic on the basis of estimated average requirements, the PVC net can take advantage of the on-off statistics of voice traffic as well as the high peak-to-average ratio characteristic of data traffic such as that associated with interactive terminals. Other data traffic, such as file transfer, which have peak-to-average ratios closer to unity would be scheduled appropriately.

Clearly, the use of average rates in scheduling traffic will be most effective in situations when the law of large numbers will apply. For speech, experience shows that link capacity on the order of 50 to 100 times the peak rate needed for a single conversation is sufficiently large to allow effective use of average rates in scheduling traffic. For interactive data, there does not appear to be any documented evidence, but the success of packet nets in handling low-speed terminals suggests that averages will work for this class as well, although the variance is likely to be much larger in relation to the mean than is observed for speech.

To help insure that the law of large numbers will apply, the PVC routing process should accept connections only when the requested average data rate is at most a small fraction (say 2 percent) of the link capacity. It should also limit the total capacity allocated to file transfer connections having a low peak-to-average ratio in order to avoid reducing the fraction of the link capacity available for statistical averaging. In addition, the routing process must use appropriate safety margins to avoid disastrous overloads. Simulations have been planned to test the effectiveness of routing procedures meeting these criteria in realizing efficient use of network resources.

A feature of packet networks which comes about implicitly from their policy of routing packets individually is their ability to reroute traffic around defective links and nodes in the net. In the PVC net with its connection routing policy, the process of rerouting is somewhat more cumbersome but poses no conceptual difficulties. However, it may well be the case, particularly under heavily loaded conditions, that it would not be possible to reroute all connections because the reconfigured net would lack sufficient capacity. Current PVC plans do not include rerouting capability built into the routing process itself. Rather, the process, as in circuit-switched nets, could be handled by the subscribers who would request reconnection in the event of a network failure.

d. Flow Control

The routing mechanisms in the PVC net using average rate requirements and statistically obtained safety margins are intended to control network traffic flows so that the probability of overload at any point is kept small. However, that probability cannot be made equal to zero,

and therefore other mechanisms are required to deal with overloads when they occur. Three such mechanisms are proposed. One is simply to discard packets when queues become excessive. For voice traffic, experience shows that this process would not materially damage speech quality if the frequency at which it had to be invoked was not too high.[3] For data traffic, discarded packets are likely to be more damaging, and retransmission would probably be required. Since there is always the probability of packet loss due to line errors, a mechanism for retransmissions of data packets would be needed in any case, and the principal cost of discarded packets would be a reduction in effective throughput.

A second mechanism would protect against overloads by limiting the peak data rate generated by a subscriber to the value agreed upon when the virtual circuit was set up. This mechanism would operate at the periphery of the net and hold off excess traffic at the source. The third mechanism would monitor average data rates on connections using a moving window of a length appropriate to the type of connection. This mechanism would protect the network against voice encoders whose speech activity detectors were malfunctioning, as well as data subscribers who might set up a keyboard terminal virtual connection and then plug in a tape device capable of sending data at much higher average rate than a human typist. The response of the third mechanism could be either to hold off the offending source or to adjust the routing parameters for the connection to properly reflect the actual average rates. Feedback from these overload protection mechanisms to the routing process would allow the network to adapt its scheduling policies and safety margins to the load conditions actually being experienced.

e. Data in the PVC Network

The PVC net departs in many ways from a pure packet-switched network. The departures have been motivated for the most part by a desire to handle voice traffic in a satisfactory manner. Since the net lacks the hop-to-hop and end-to-end acknowledgments common in packet nets and may actually discard packets under overload conditions, there is some non-negligible probability that packets will fail to arrive at the destination node. It is assumed that most data subscribers would require guaranteed delivery of data packets. End-to-end protocols have not yet been worked out, but would most likely be similar in character to the windowed acknowledgment scheme discussed by Cerf and Kahn for internetting applications.[6] The processing and buffering necessary to implement the data protocol would exist at the periphery of the PVC net, and could be provided either by the network proprietors or by the data subscribers. The overhead traffic associated with acknowledgments and retransmissions has not yet been taken into account in our simulations.

f. Supervisory Traffic in the PVC Network

The process of setting up and taking down calls in the PVC net would be effected by sending supervisory packets from node to node through the network. These packets would flow on predefined virtual connections between all pairs of adjacent nodes, providing a path for messages of arbitrary length between nodes. The packet streams on these connections would follow a protocol similar to the data protocol to effect guaranteed delivery. Supervisory packets would be intended for adjacent nodes, and thus no forwarding would be performed. Some supervisory processes would, of course, result in a sequence of supervisory messages propagating through the net, e.g., setting up a call.

2. Model Description

To investigate the viability of PVC techniques, a model of a single link of a PVC network has been developed and simulated on the PDP 11/45 computer.[7] The simulation models a population of speakers in conversation, providing a voice load on the system. Data traffic is modeled by a Poisson process. The PVC link model permits the investigation of such variables as buffer space requirements, packet delay, and line utilization as functions of the voice and data loads on the system.

As shown in Fig. IV-1, a single link is modeled as having two distinct input queues for voice and data traffic. When the link is available, a packet is chosen from one queue or the other and transmitted. A summary of the fixed parameters of the model is presented in Table IV-1. The maximum size of the data queue is a variable that is measured for different traffic loads.

TABLE IV-1

FIXED PARAMETERS IN THE LINK MODEL

Packet size	128 bits
Overhead in packet	32 bits
Data in packet	96 bits
Channel rate	1.544 Mbps
Supervisory traffic and framing	8 kbps
Available channel rate	1.536 Mbps 12,000 packets/sec
Vocoding techniques	CVSD, LPC
CVSD vocoding rate	16 kbps 6 msec between packets
LPC vocoding rate	3.5 kbps 27.5 msec between packets
Voice queue size	70 packets 560 16-bit words 5.83 msec of channel time
Simulation duration	2 min. of channel time

For each run of the simulation, the total number of speakers for each of the vocoders is set. Each speaker is determined to be speaking (active) or silent according to distributions of talkspurt and silence distributions obtained from measurements by Brady.[8] When a speaker is determined to be active, he generates packets at a rate characteristic of the vocoding technique. When silent, no packets are generated. The model does not attempt to represent the start or end of conversations. When a voice packet is generated, it is entered into the voice queue for transmission. The voice queue is finite; when it is filled, half the packets in the queue are discarded.

Some of the simulation variables are displayed on a color CRT unit. The display can be printed with a rough color coding. Such an output is shown in Fig. IV-2 where the number of active speakers, and the maximum and minimum size of the voice queue are plotted vs time. Each point on the curves represents the average of the particular variable over 6 msec.

Speech loss events are shown as vertical bars. The data in Fig. IV-2 result from a voice load of 135 CVSD speakers and no data load.

C. BEHAVIOR OF VOICE AND DATA TRAFFIC IN THE PVC LINK SIMULATION

1. Voice Queue

Initial measurements on the simulation were made with only voice traffic. It was assumed that voice had absolute priority on the link; measurements were collected on the remaining, unused portion of the channel to predict the behavior of the data channel (see Sec. 2 below). Table IV-2 shows some typical results from simulation runs. In the first column are listed the total number of speakers for the particular run, and ρ_v is the utilization of the channel capacity for voice traffic. $\overline{X}$ and σ_x are measurements on the unused fraction of channel capacity. $\overline{X}$ is the mean duration of contiguous voice slots (packet times) or mean time between empty packet slots, while σ_x is the standard deviation. If all empty slots were used for data, the amount of data traffic R_D that could be transmitted over the channel is listed in the last column.

TABLE IV-2

VOICE CHANNEL PERFORMANCE

	ρ_v (percent)	$\overline{X}$ (μsec)	σ_x (μsec)	Max R_D (kbps)
130 CVSD	90.0	834.3	15,037.8	115.066
120 CVSD	81.8	456.8	4,697.0	210.159
110 CVSD	76.5	354.8	645.0	270.538
100 CVSD	70.6	283.6	483.4	338.537
100 CVSD, 135 LPC	90.7	894.3	10,253.9	107.348
75 CVSD, 110 LPC	69.4	272.6	469.71	352.124
75 CVSD	51.4	171.6	239.9	559.302

Loss of voice packets due to overflow of the voice queue was observed to occur in those cases where ρ_v was greater than 80 percent. However, only for the 130 CVSD speaker case did the channel lose a significant fraction of the packets (~13.5 percent) for a long enough interval (~3.5 sec) to be objectionable.

2. Data Queue

a. Data Channel Performance Determined from Measurements of the Voice Queue

(1) Assumptions and Predictions of an M/G/1 Queue

Given the statistics of the unused fraction of the channel, one can attempt to predict the behavior of the data queue by modeling the data queue as an M/G/1 queue.[7] The gaps between

empty slots can be considered as service times for data packets. The use of the M/G/1 queue model requires the following assumptions:

(a) The data packets arrive according to a Poisson model,

(b) Voice traffic has absolute priority, and

(c) Successive service times (gaps between non-voice slots) are independent.

The first two assumptions are clearly met in the model; the third assumption is questionable and will be discussed below.

The M/G/1 model provides[9] a formula for the mean waiting time $\overline{W}$ for a packet in the data queue as a function of data arrival rate. Some theoretical curves are plotted in Fig. IV-3. These curves permit the estimation of link utilization as a function of the desired mean waiting time for data packet transmission. For example, if one chooses a population of 100 CVSD speakers, a load just slightly greater (104 percent) than the trunk could handle with pure circuit-switching, the predicted mean wait is 8.3 msec when transmitting 320 kbps of user data (427 kbps when packet overhead is included). Under these conditions, no voice packets are lost and the worst-case delay for voice is less than 6 msec. Voice traffic is using 70.6 percent of the packet slots, and the queueing model predicts that data occupy 94.5 percent of the remaining slots. Total channel utilization is 98.4 percent. If packet overhead is considered, the net utilization for voice and data is 73.8 percent of link capacity.

(2) Independence of Successive Data Packet Service Times

The suspect assumption for using the M/G/1 queue model — independence of successive service times — was checked by computing the first serial correlation coefficient of the duration of successive contiguous voice packet intervals. Following Cox and Lewis,[10] $\tilde{\rho}_1$ is the unbiased estimate of the first serial correlation coefficient:

$$\tilde{\rho}_1 = \frac{\sum_{i=1}^{n-1} \left[x_i - \frac{1}{n-1}\sum_{i=1}^{n-1} x_i\right]\left[x_{i+1} - \frac{1}{n-1}\sum_{i=1}^{n-1} x_i\right]}{\sum_{i=1}^{n-1}\left[x_i - \frac{1}{n-1}\sum_{i=1}^{n-1} x_i\right]^2} \qquad \text{(IV-1)}$$

The quantity $\tilde{\rho}_1 \sqrt{n-1}$ will have a unit normal distribution if ρ_1, the actual correlation coefficient, is zero and n is large. Independence is rejected as a hypothesis at the α significance level if

$$|\tilde{\rho}| > \frac{C_{1/2\alpha}}{n-1} \qquad \text{(IV-2)}$$

where $C_{1/2\alpha}$ is the upper $1/2\alpha$ point of the unit normal distribution.

The correlation measurements are summarized in Table IV-3. The successive intervals between empty slots fail the test for independence at significance levels of 5, 2, and 1 percent. It follows then that the M/G/1 queue model may not provide accurate predictions for the behavior of the data queue. Nonetheless, it may provide some bound or approximation to the queue behavior. Further investigations of independence are planned. The queueing theory predictions and simulation measurements are compared in the next section.

TABLE IV-3

COMPUTATIONS OF FIRST SERIAL CORRELATION COEFFICIENT
100 CVSD SPEAKERS
($n = 65,536$)

Estimates of Coefficient $\tilde{\rho}_1$
−0.051759
−0.054166
−0.023981
−0.036636
−0.033078
−0.013363

Significance (α) Level (percent)	$C_{1/2\alpha}/\sqrt{n-1}$
5	0.007656
2	0.009102
1	0.010060

b. Experimental Measurements on the Data Queue

The basic measurement for determining the behavior of data traffic is a histogram indicating the number of occurrences of each length (number of packets) of the data queue. The average waiting can be computed using Little's result[11]

$$\overline{W} = N_q/\lambda \tag{IV-3}$$

where N_q is the average number of packets in the queue, and λ is the average arrival rate of packets per slot time. N_q can be calculated from Eq. (IV-4) where

$$N_q = \frac{\Sigma i \cdot n_i}{N} \tag{IV-4}$$

where i is the length of the data queue in packets, n_i is the number of slot times which that length queue occurred during the simulation run, and N is the total number of slot times during the run. Recognizing that λ is the average number of data packets transmitted during the simulation run, the mean waiting time becomes

$$\overline{W} = \frac{\Sigma i \cdot n_i}{n_d} \tag{IV-5}$$

where n_d is the total number of data packets transmitted. The measured waiting time* is compared with that predicted by the M/G/1 queue in Fig. IV-4. The results from the simulation

* When the PVC link is run, a software random number generator produces the sequence of numbers used to select talkspurt and silence durations from their respective distributions. Usually, the number generator is started with a random "seed" for each run of the simulation. However, when computing data points from many simulations for a particular curve or family of curves, the same "seed" is used. The specific effects of one variable (for example, input rate of data traffic) can thus be analyzed without the dispersion in results due to run-to-run variations in speaker activity.

indicate that the fraction of the channel not used by voice traffic cannot carry as much data at the same delay as that predicted by the M/G/1 queue. For a 6-msec delay, the M/G/1 queue predicts a data load of approximately 450 kbps and total link utilization of 99 percent, while the simulation measurements indicate a data load of about 330 kbps and total link utilization of 90 percent.

In summary, the measurements indicate that although, in addition to voice traffic, there is sufficient net capacity available for a rate of data traffic that brings the total link utilization to 98-99 percent, the statistics of voice traffic with absolute priority are such that very large delays in data packet transmission and unacceptably large queue lengths result. Nonetheless, one can maintain acceptable delays and queue lengths when data traffic is introduced at rates that result in net link utilizations of 90-92 percent – still relatively high values.

D. ADDITIONAL MEASUREMENTS ON A SINGLE PVC LINK

1. Varying Data and Voice Priorities

When data traffic was introduced into the simulation, the color display routines were modified to display the behavior of the data queue as well as the voice queue. When voice traffic has absolute priority, data packets are transmitted only when the voice queue is empty. It is observed on the display that often the voice queue remains small, but nonzero, for extended periods of time; thus, data transmission is delayed and, despite the fact that the average data rate is close to anticipated values, a very large data queue results. Consequently, different priority strategies for transmitting voice and data were investigated.

A framing strategy can be introduced in which a fraction of the packet slots in the frame have priority for data. The fraction can be made to vary according to the voice and data loads in the node.

Figure IV-5 presents the same data as Fig. IV-4 with the addition of measurements made when voice packets had priority for only 7 out of every 10 packets. The change in priority decreases the mean wait for data and the maximum size of the data queue, and increases somewhat the delay for voice packets. For the voice load depicted in Fig. IV-5, the decrease in the number of packet slots with voice priority did not increase voice packet delay significantly enough to result in speech loss. However, in cases with larger voice loads which result in no speech loss with absolute voice priority, there is speech lost when some priority is given to data.

A more detailed picture of the effects of changing voice and data packet priorities is shown in Fig. IV-6 where the data are all plotted against the number of packets (out of 10) with voice priority. The data and voice loads on the link are exceptionally heavy and would not be used in a practical situation. Such a traffic load provides a large dynamic range of speech loss and data waiting time such that the effects of varying the voice/data priorities may be observed. The speaker load can potentially utilize about 80 percent of the link capacity, while the data load can potentially utilize about 30 percent. When voice has zero priority, voice packets are transmitted only when the data queue is empty and approximately 11 percent of the speech is lost (when the voice queue overflows, half its packets are discarded). This speech loss does not decrease substantially until 80 percent of the packet slots have priority for speech. When speech has all or nearly all the priority, it utilizes nearly 80 percent of the link capacity without any queue overflow. Data packets fill in the remaining packet slots, resulting in a net link

utilization of almost 100 percent. However, since more data packets are presented to the link than can be handled, queued data packets build up indefinitely.

2. Techniques of Discarding Packets and the Size of the Voice Queue

When the simulation was initially constructed, it was arbitrarily decided to discard half the queue (35 packets) when there was an overflow. An appropriate question was, for a heavy traffic load can the amount of speech lost be decreased if fewer packets are discarded during overflows and/or if the size of the voice queue is increased (increasing the maximum delay for speech packets). The answer to the question is shown in Fig. IV-7. For a given set of voice and data traffic loads and priorities, the percentage of speech lost is plotted against the size of the voice queue. There are two curves: one in which half the queue is discarded at overflows, and the other in which only incoming packets are discarded when the queue is full. Several results can be discerned from the graph. Speech loss cannot be significantly decreased by increasing the size of the voice queue. The increase in maximum packet delay probably outweighs the small decrease in loss. It also appears that voice queue overflows occur in bursts, since discarding only incoming packets does not save appreciably more speech than discarding 35 packets when the queue reaches capacity. The optimal queue size is then between 50 and 100 packets (4.167 to 8.33 msec maximum delay), and only incoming packets ought to be discarded when the voice queue is full.

3. Behavior of Smaller-Capacity Links

The original PVC link simulation assumes a link capacity of 1.544 Mbps. When speech activity detectors are used and no voice packets are transmitted during silence, a 1.544-Mbps link can handle approximately 100 to 125 16-kbps CVSD speakers with minimal speech loss. Experience[12] shows that the TASI advantage can safely be used only when the capacity of the channel shared by the conversations is relatively large (the order of 50 to 100 conversants). How does the PVC concept fare in networks with smaller capacity links? More specifically, given the smaller capacity links that cannot benefit from the TASI advantage, can the remaining capacity (unused by voice) be used (with acceptable delays) by data? These questions were investigated, and the results follow.

Figure IV-8 plots the percentage of speech lost against the utilization of the channel for voice for several link capacities. No data were transmitted. Clearly, as is predicted in the literature,[12] smaller-capacity links cannot use the TASI advantage as well as those with larger capacity. At a speech loss level of 1 percent, 93 percent of a 1.544-Mbps channel is utilized for voice traffic, while only 73 percent of a 128.64-kbps channel is utilized for voice traffic.

A 0.1-percent speech loss level was selected and voice loads were determined from the curves in Fig. IV-8 for the 128.640-kbps and 1.544-Mbps links. For the smaller-capacity link, 7 speakers result in a link utilization of 0.59; for the larger-capacity link, 123 speakers result in a link utilization of 0.87.

The PVC link simulation was run at these voice loads with varying data loads; the data rates were restricted by requiring acceptable mean waiting times and queue lengths. The utilization of the link by data and the mean waiting time for data packets were measured. The results are plotted in Fig. IV-9. Link utilizations (ρ's) are plotted on a linear scale; mean waiting times ($\overline{W}$'s) are plotted on a log scale. Simulations were first run with absolute priority of voice over data. For the 128.640-kbps link, utilization of the link for data ranged from 0.052 to 0.312 with

mean waiting times ranging from 54.4 to 1027.1 msec. The maximum length of the data queue varied from 366 to 3434 data packets. For the 1.544-Mbps link, the data utilization ranged from 0.022 to 0.087 with mean waiting times ranging from 35.8 to 641.7 msec. The length of the data queue varied from 310 to 4834 data packets.

The utilization of the link by data in the smaller link is 2 to 4 times that of the larger-capacity link, but the net utilization (including packet overhead) is still only 0.689 while that of the larger link is 0.913.

Further simulations were run with some priority given to data packets. In the 128.640-kbps link, voice was only given priority for 70 percent of the slots ($\rho_V = 0.59$); in the 1.544-Mbps link, voice was given priority for 90 percent of the slots ($\rho_V = 0.87$). The results of these runs are indicated by dashed curves in Fig. IV-9. The changes in priority increase speech loss to 5.0 percent in the smaller-capacity link and 1.4 percent in the larger. Mean waiting time decreases significantly, but utilization of the links by data does not increase.

One can conclude that the high link utilizations by voice and data which result from PVC techniques are not achieved with significantly lower-capacity links. Although the lower-capacity link can carry proportionally more data, the utilization of the link by voice dominates the overall channel utilization for the given transmission priorities.

REFERENCES

1. J. W. Forgie and A. G. Nemeth, "An Efficient Packetized Voice/Data Network Using Statistical Flow Control," Proc. National Computer Conf., Chicago, IL, 12-15 June 1977.
2. J. W. Forgie, "Speech Transmission in Packet-Switched Store-and-Forward Networks," Proc. National Computer Conf., Anaheim, CA, 19-23 May 1975, pp. 137-142.
3. ____________, "Subjective Effects of Anomalies in Packetized Speech," J. Acoust. Soc. Am. 60, Suppl. 1, 5109 (1976) (Abstract).
4. D. W. Davies and D. L. A. Barber, Communications Networks for Computers (Wiley, London, 1973), p. 361.
5. L. Kleinrock, W. E. Naylor, and H. Opderbeck, "A Study of Line Overhead in the ARPANET," Commun. ACM 19, 3-12 (1976).
6. V. G. Cerf and R. E. Kahn, "A Protocol for Packet Network Intercommunication," IEEE Trans. Commun. COM-22, 637-648 (1974).
7. A. G. Nemeth, "Behavior of a Link in a PVC Network," Technical Note 1976-45, Lincoln Laboratory, M.I.T. (7 December 1976), DDC AD-A036370.
8. P. T. Brady, "A Technique for Investigating On-Off Patterns of Speech," Bell Syst. Tech. J. 44, 1-27 (1965).
9. L. Kleinrock, Queueing Systems, Vol. 2: Computer Applications (Wiley, New York, 1976), pp. 16-19.
10. D. R. Cox and P. A. W. Lewis, The Statistical Analysis of Series of Events (Methuen, London, 1966).
11. L. Kleinrock, Queueing Systems, Vol. 1: Theory (Wiley, New York, 1975), p. 17.
12. K. Bullington and J. M. Fraser, "Engineering Aspects of TASI," Bell Syst. Tech. J. 38, 353-364 (1959).

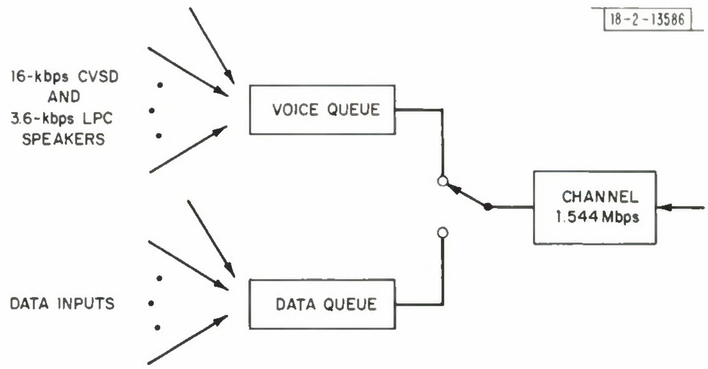

Fig. IV-1. Model of single link in PVC network.

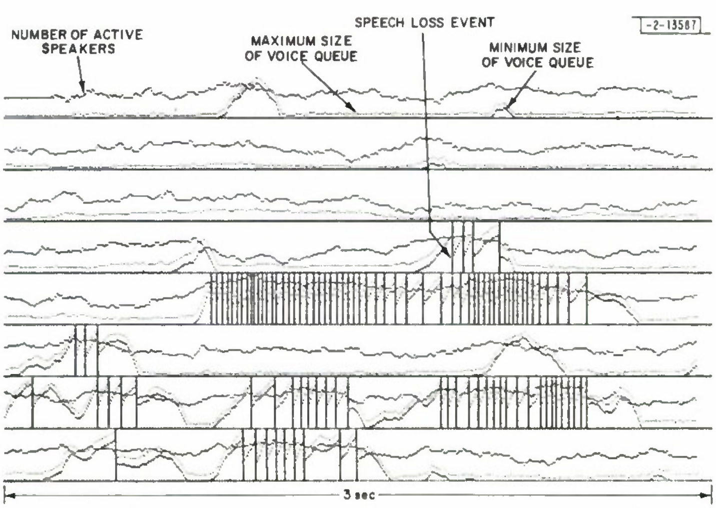

Fig. IV-2. 24 sec of output from PVC link simulation.

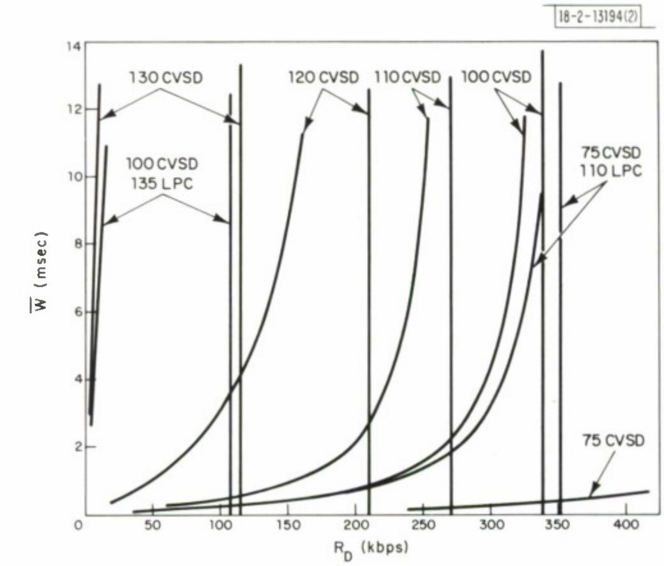

Fig. IV-3. Mean waiting time vs data rate. Vertical bars are maximum data rate. Levels of voice load are shown.

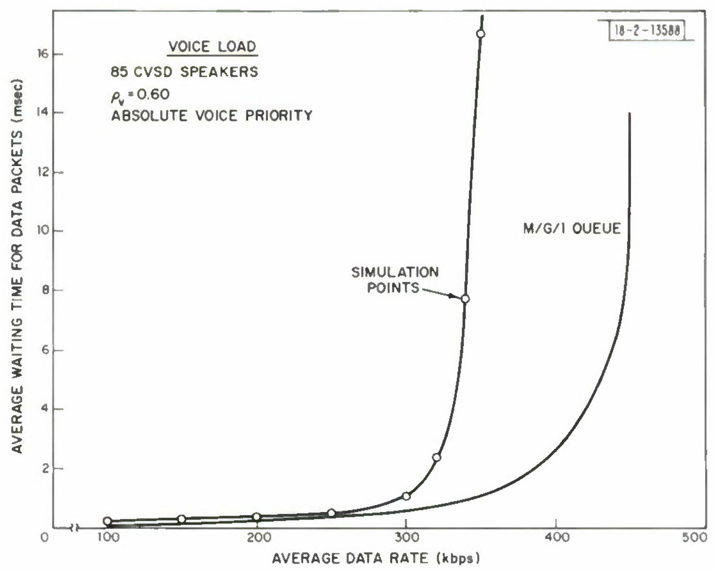

Fig. IV-4. Average waiting time vs data rate.

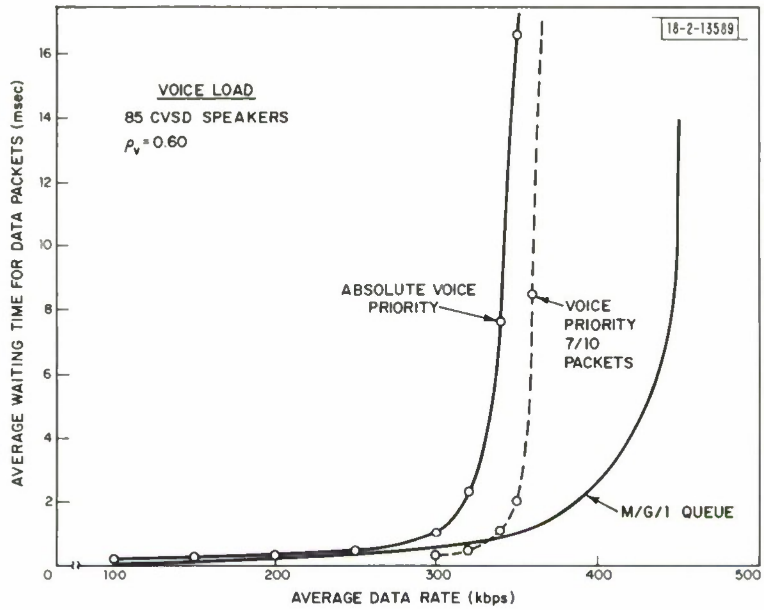

Fig. IV-5. Average waiting time vs data rate.

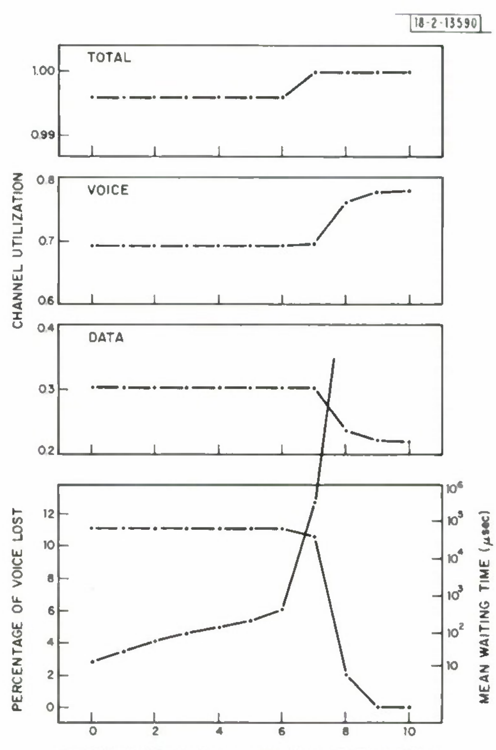

Fig. IV-6. Voice and data priority.

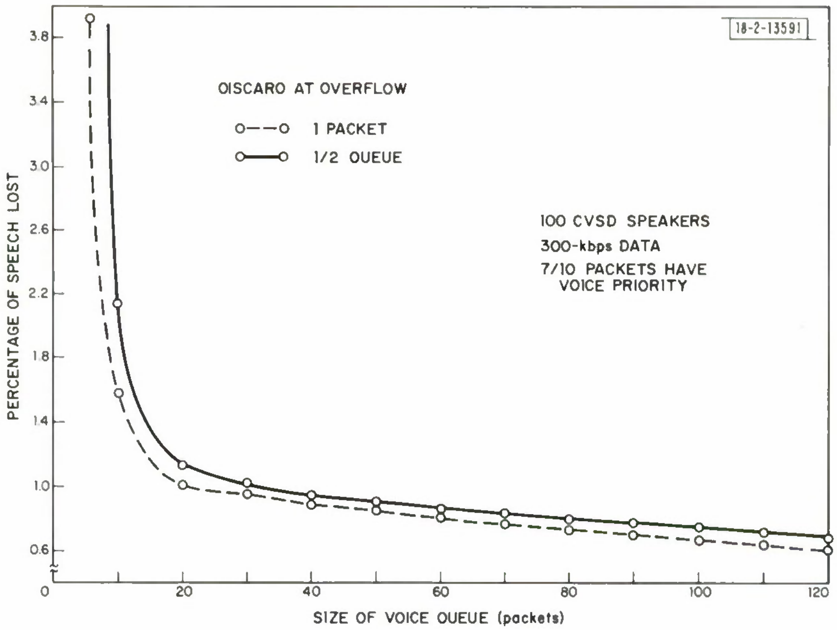

Fig. IV-7. Speech loss vs size of voice queue.

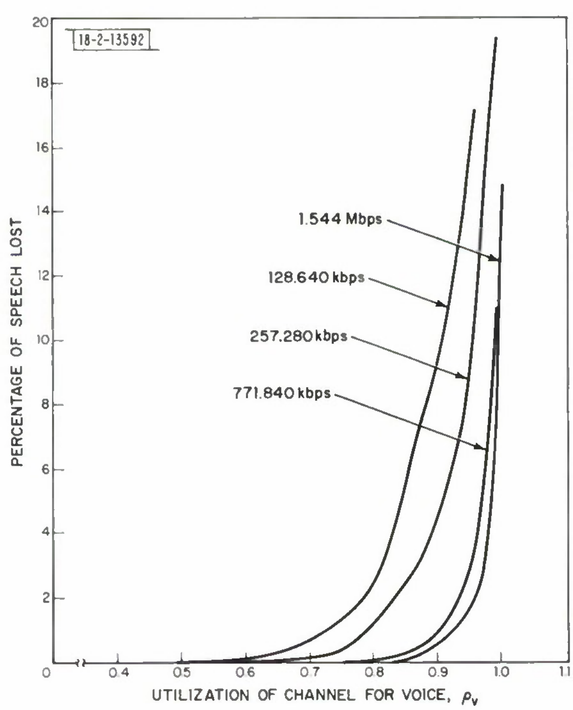

Fig. IV-8. Speech loss vs utilization for different link capacities.

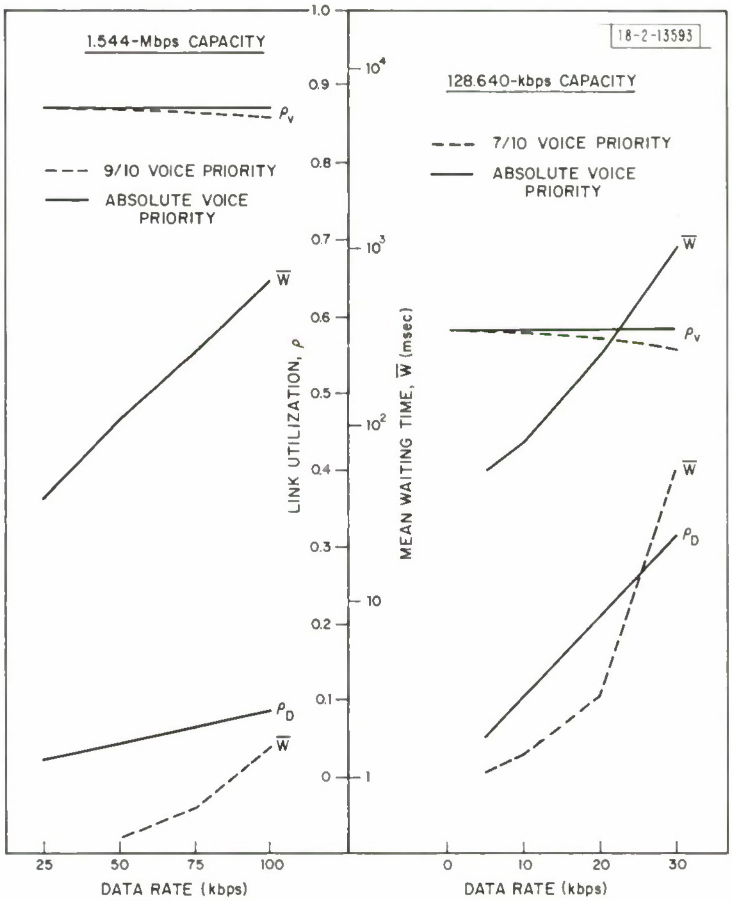

Fig. IV-9. Link utilizations for voice and data, and mean waiting times for data packets.

APPENDIX A
THE SECURE VOICE CONFERENCING FACILITY

I. INTRODUCTION

Part of Lincoln Laboratory's ongoing commitment to narrowband speech and data network design involves various speech conferencing studies. Under our current program for the Defense Communications Agency, we are planning to simulate and evaluate present and proposed conferencing bridge arrangements and demonstrate one or more of the most attractive techniques. In this appendix, we discuss a very flexible signaling and switching arrangement configured around the PDP 11/45 and capable of implementing all the high- and low-rate conferencing schemes now postulated. The proposed system includes sufficient flexibility to allow for most audio, delta modulation, or frame bridging proposals. With the aid of external voice digitizers, easily connected to the system when required, combinations of conferenced and tandemed speech signals can be processed.

Basically, the system consists of two independent sections – a control section and an audio conditioning section. The control section is composed of 20 touch-tone data sets connected to dial-up Bell system lines. These lines are automatically answered to establish a user to computer connection, and then are used to transmit touch-tone commands from a user to a PDP 11/45. These commands control conference configurations and conference queues in real time. The audio conditioning section consists of a multiplexed A/D-D/A system and a large buffer memory connected to a signal processing machine (a Lincoln Laboratory DVT). This machine allows audio connections to be made arbitrarily between users. In addition, three ports on the A/D-D/A system are to be used for external voice equipment. The large buffer memory can implement delays of up to 0.5 sec for each of the 20 dial-up users. For additional flexibility, the signal processor is also connected to the 11/45 so that the control inputs can be used to modify the switching and signal processing operations in real time.

II. SYSTEM DESCRIPTION

Figure A-1 is a block diagram of the complete conferencing facility. From the point of view of the PDP 11/45 machine, two external devices are connected through standard DEC interface circuits. The telephone control system is connected through a standard DR11C single-word interchange board with interrupt capability. The audio-switching section is connected through a more flexible DR11B direct memory access (DMA) interface. Twenty 2-wire phone lines are connected to the touch-tone receivers for the control path and a set of hybrid (2- to 4-wire) transformers for the audio path. The 4 wires from each of the 20 lines are connected to an A/D-D/A converter port for audio switching.

A. The Touch-Tone Receiver Control Path

Each of the 20 phone lines will be connected to a Bell type 403 tone data set which automatically responds to a ringing signal by passing a ringing bit (R) to the computer interface. If the computer raises a data terminal ready bit (DTR), the data set will answer the line and set up to receive control tones by transmitting a data set ready (DSR) bit. When a user presses a tone button, the data set will signal the computer with a data carrier detector (DCD) bit, and a 4-bit tone code. The computer can listen for these tones, have the data set transmit three single-frequency responses, or hang up.

Figure A-2 presents the interface between 20 data sets and the DR11C. The basic interface function scans the 20 data sets for activity by comparing a new status word from each channel with a previous stored status word for the same channel. Each previous channel status word has been stored in the 32- × 4-bit RAM. Only the three status bits (R, DCD, and DSR) need be stored for comparison against the latest word. If there is a change in any of these bits where change is defined as: $DCD \cdot \overline{DCD}_{-1} + R \cdot \overline{R}_{-1} + DSR \otimes DSR_{-1}$, then the present word, including a 5-bit code for channel identification, is clocked into a first-in/first-out (FIFO) buffer and an output request is set for the computer to inspect or be interrupted. The 20 data sets are scanned in a cycle of 20 of the 8-kHz (125-μsec) samples (see Fig. A-3), so that a complete scan requires 20 × 125 × μsec = 2.5 msec. Each data set is controlled from the interface by a 5-bit register which is loaded under program control from the PDP 11/45 – DR11C path.

B. The Audio Conditioning Section

As Fig. A-1 indicates, the audio conditioning section consists of three subsections: a Lincoln Laboratory DVT signal processing computer, a multiplexed A/D-D/A system which is controlled by and communicates with the DVT, and finally a large (160K) core memory which is controlled from the DVT. The DVT, in turn, can also communicate with the PDP 11/45 through a DMA interface called a DR11B.

1. The Multiplexed A/D-D/A System

The A/D-D/A system is shown in Fig. A-4. It is connected to the channel 0 input and output ports of the DVT and consists of an A/D section, a D/A section, and some multiplexing timing registers.

The A/D section can accept up to 32 input analog signals multiplexed through two Teledyne 16:1 gates (only 23 inputs will be used). These multiplexer gates drive a sample-and-hold (S/H) gate which drives, in turn, a 12-bit A/D converter. The multiplexed input is controlled from a 5-bit register incrementer which can be loaded with a 5-bit word asynchronously so that random access conversion of any input channel can take place; or, a standard input clock will increment the register by one each cycle and clear at some settable value. In other words, the input multiplexer can be stepped randomly, or cycled through a fixed pattern. A normal input rate will be 200-kHz (5-μsec) conversions, although an external clock can be used. The input A/D 12-bit word will be read on input channel 0 of the DVT, either as a forced input or an interrupt.

The D/A section is double buffered, which means that the user can load the D/A buffer on a channel 0 output from the DVT but the transition of the D/A converter will take place on the next synchronous clock edge. A demultiplexer S/H gate will be controlled by a 5-bit word delayed by one clock cycle from the input MUX control. This allows for the delay in D/A conversion. The D/A section consists of the double buffering, a fast 12-bit D/A converter, a set of 23 (expandable to 32) S/H gates, and a 5-bit decoder-pulse steerer. The choice of S/H outputs rather than individual slower D/A registers and converters was based on cost and wiring complexity.

2. The Large Buffer Memory and Interface

Basically, the large buffer memory to be used in the conference system is a 128K by 20-bit core memory plus a 32K by 20-bit core memory, both have about a 2-μsec read-modify-write. We are designing a 16-bit word interface, since that is the DVT data word width. In fact, our

delay experiments will mostly require only 12-bit words. The input to and output from the memory (write and read words) will be communicated from and to channel 2 of the DVT. Actual read, write, read-modify-write, load address, and various hybrid commands to the large memory will be transmitted from output channel 0 of the DVT. Since this channel was designed as a 12-bit output to a D/A converter, 4 more bits are available to be decoded and steer data to other places besides the D/A converter. The lower-left portion of Fig. A-5, the memory interface and channel 0 decoder, shows the decoding table. An output on channel 0 from the DVT with 4 upper bits zero produces a standard D/A load. The other commands load upper and lower portions of the 18-bit address register, and start read, write, or read-modify-write cycles. Since the output on channel 0 is a 12-bit word, the loading of the address register is a two-command operation. The lower address (A_L) is 12 bits and the upper portion (A_U) is 6 bits. Presumably, only the lower register would be loaded for many applications requiring only one command. It is also possible to combine the address load with a read, write, or read-modify-write command. Two remaining commands set up the multiplex word and do a master clear.

3. The DVT as Controller

The DVT has a limited in-out system which will be modified to control the multiplexing system and the large memory. The present 4 channels of input and output will be assigned as follows. Channel 0 will output to the D/A converter, set the MUX index, or control the large memory as discussed. Channel 0 input will receive data from the A/D converter. Channel 1 will communicate with the PDP 11/45 through the DR11B interface. Channel 2 will deal with the large memory (M_L). Finally, channel 3 will remain as the link to M_X, the internal DVT bulk memory. The 55-nsec cycle time of the DVT allows for about 90 machine cycles during each 5-μsec A/D conversion cycle.

III. A CONFERENCE EXPERIMENT EXAMPLE

Figure A-6 shows the conferencing facility as it might be configured for a 3-party conference. This example shows a conference which is bridged at the delta modulated bit level, output to a tandem narrowband vocoder, and then distributed to the conferees.

The three participants would form the conference by dialing up one of the 20 phone numbers, and talk via touch-tone to the PDP 11/45 conference control program. The DVT software would be loaded via the 11/45 to implement CVSD encoders for each of the participants, effect the bit stream bridging, delay the audio inputs by fixed or time-variable amounts, output the decoded bridged signal to an externally connected vocoder (on channel 21, 22, or 23), and receive the output of the vocoder tandem back in on the corresponding A/D channel for distribution to all the conferees, or all except the one talking.

Figure A-6 is just an example to show the use of all the elements in the system, and especially the role the DVT software will be called upon to play.

Finally, Fig. A-7 indicates the physical layout of conferencing equipment aside from the PDP 11/45, and the large core memory used for delay.

If a fourth person wished to join the conference, he would call in and interact with the control software scanning the touch-tone interface. Then flags would be activated in the DVT to enable another A/D-D/A channel and include the fourth stream in the bridging and distribution.

For certain conferencing configurations, statistics about activity, coincidence of talkers, etc. can be gathered on-line by way of the DVT 11/45 link. Certain parameters such as delay can be fed to the DVT to simulate time-varying situations.

A second conferencing configuration is shown in Fig. A-8. This example is contrived to show the use of the extra A/D-D/A spigots beyond the 20 used for audio lines. In this case, two audio streams would be vocoded in externally connected hardware by driving the external hardware from D/A converters 21 and 22. The digital data from these units are input to the PDP 11/45 through a special-purpose "ring" interface, so that the 11/45 rather than the signal conditioning computer performs whatever frame bridging algorithm is decided upon. The result of the frame bridging algorithm is outputted through the digital interface to the synthesizer portion of a digital vocoder whose output, in turn, drives the audio conditioner at A/D input 21. The combined bridged vocoder speech could then be connected to all the conferees using the system to simulate simultaneous activity of two talkers in a vocoder bridging conference. Meanwhile, any control information fed in from the upper pathway (e.g., new conferee connected) could be delivered to the audio conditioner through its DR11B interface.

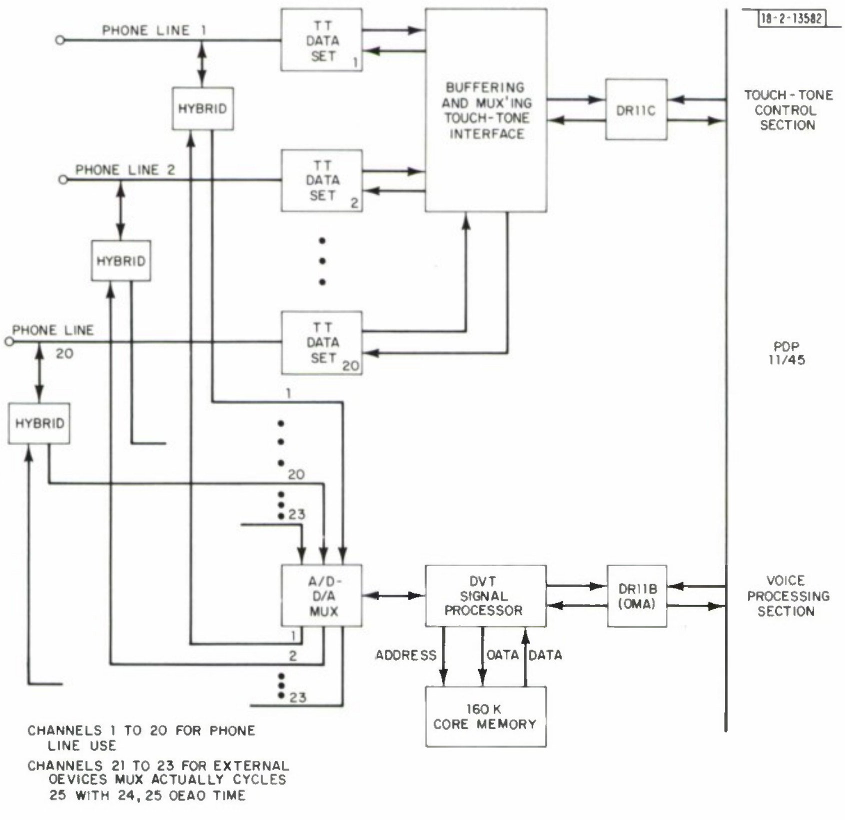

Fig. A-1. Conferencing system.

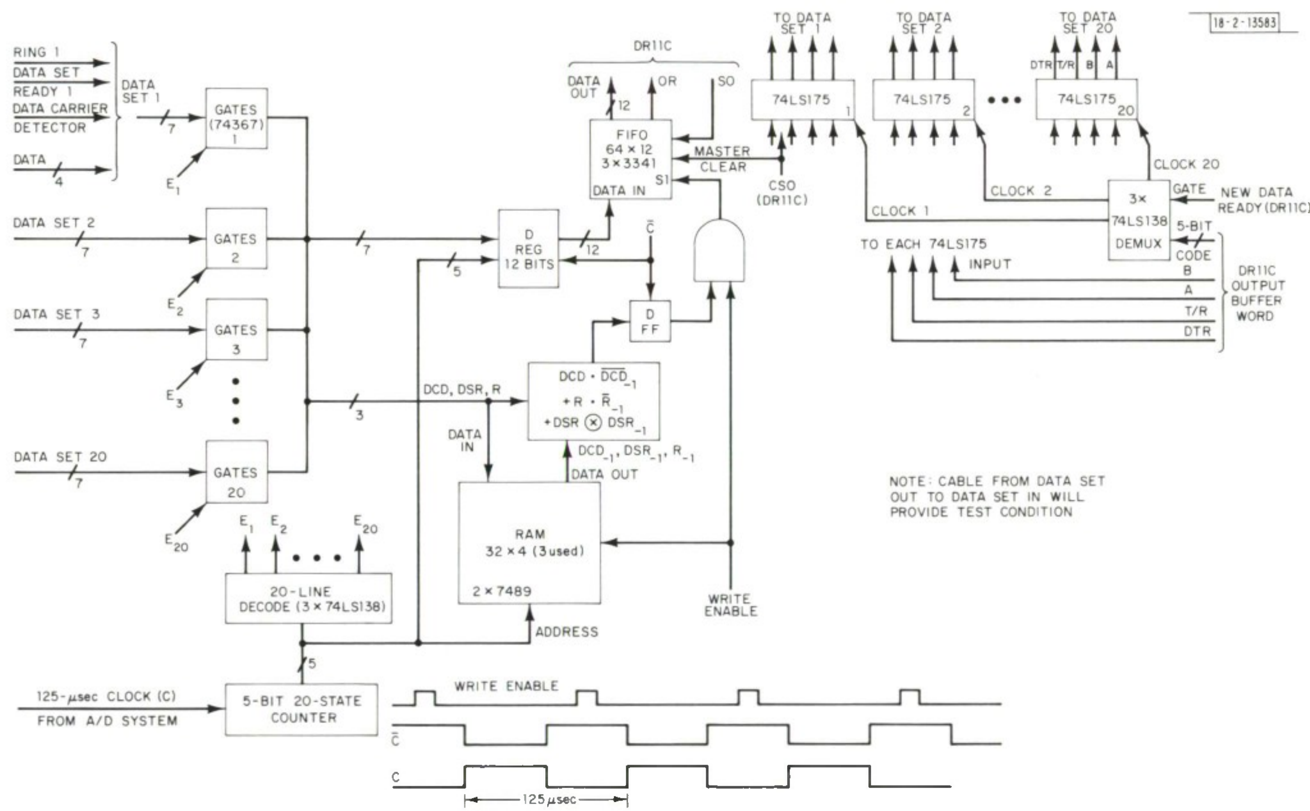

Fig. A-2. Touch-tone interface, 20 data sets to DR11C.

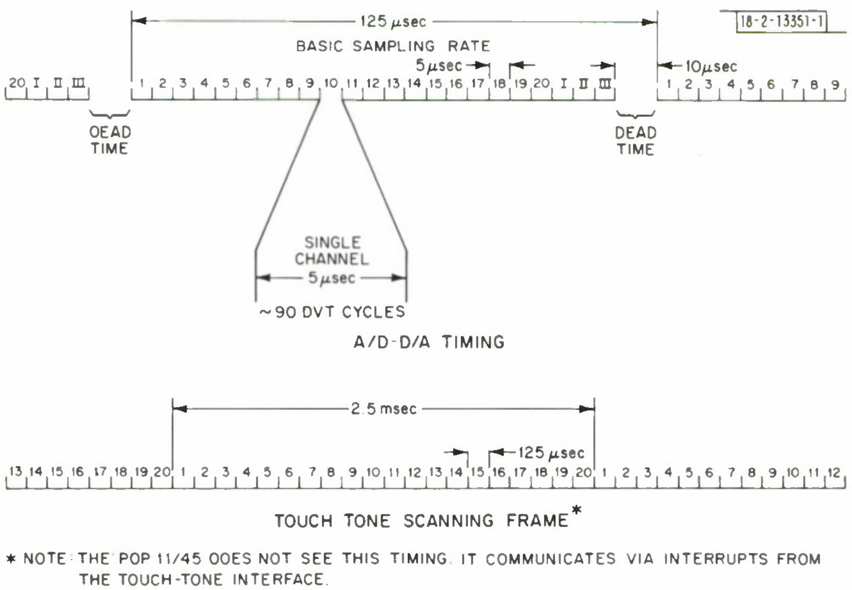

Fig. A-3. Conferencing system timing.

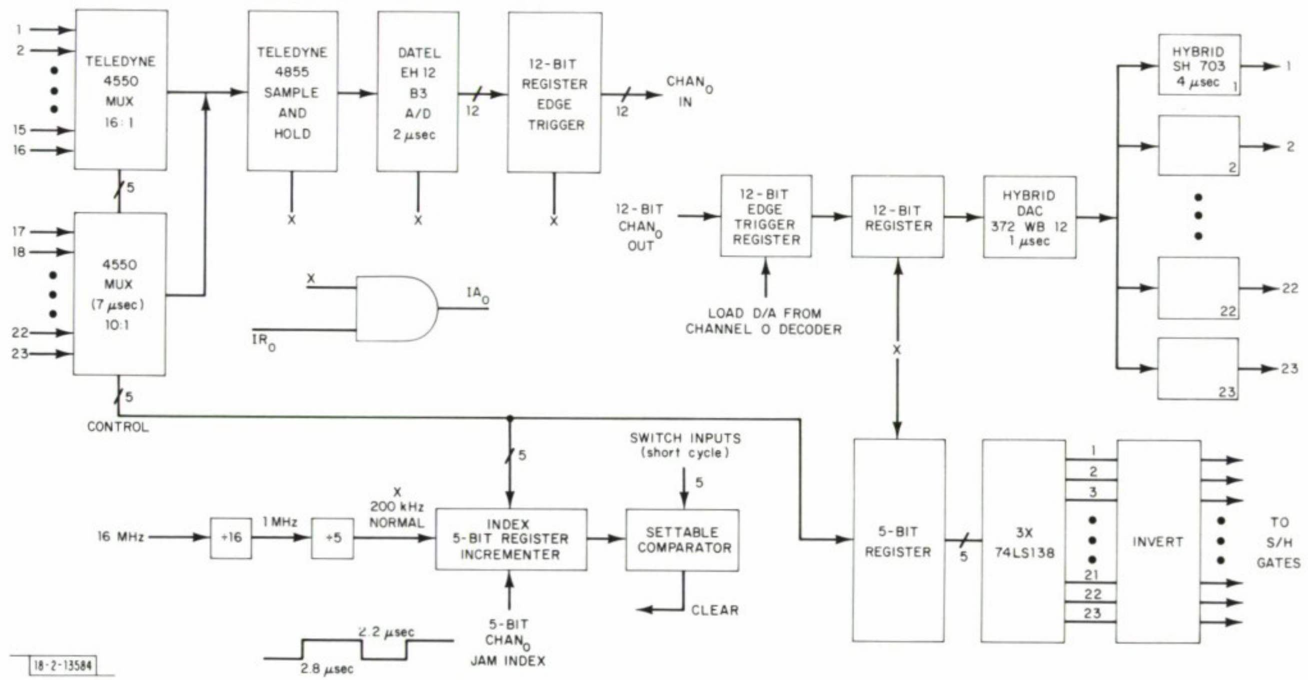

Fig. A-4. A/D-D/A-MUX system.

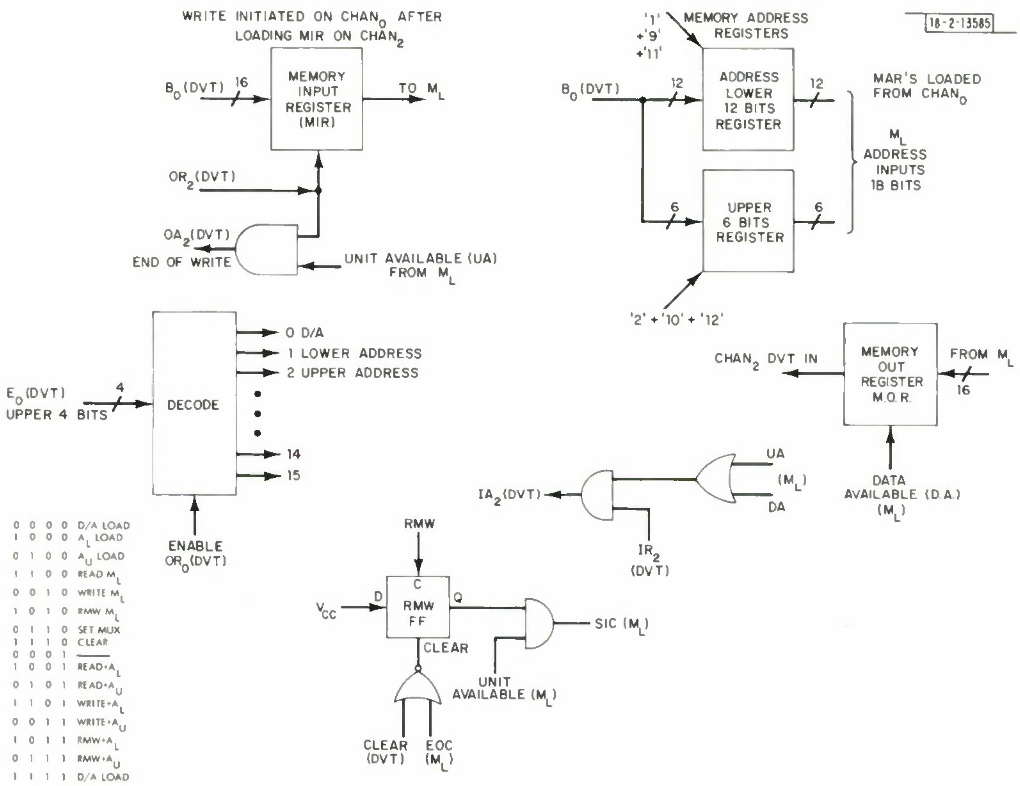

Fig. A-5. Large memory interface (M_L) channel 0 decoder.

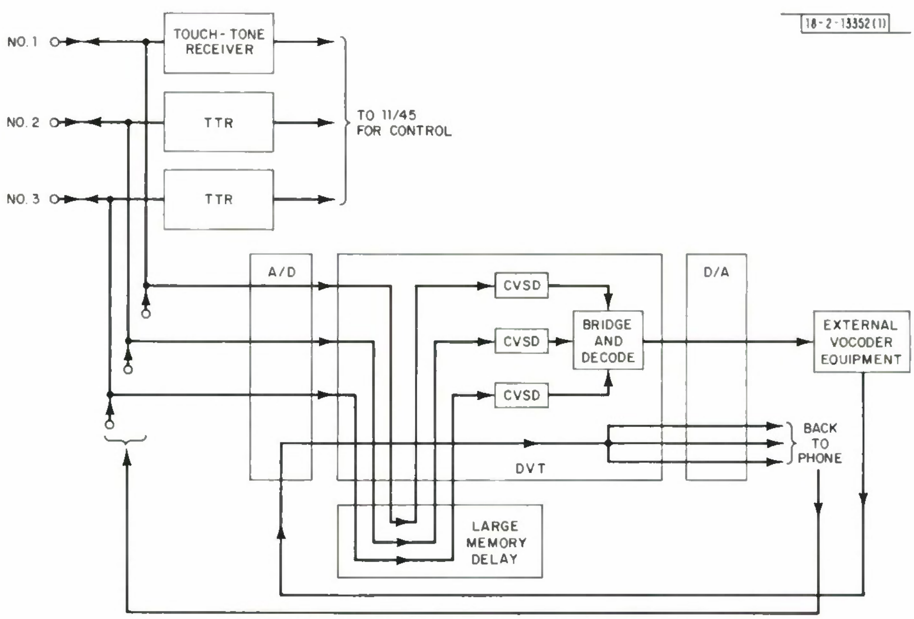

Fig. A-6. Conference example.

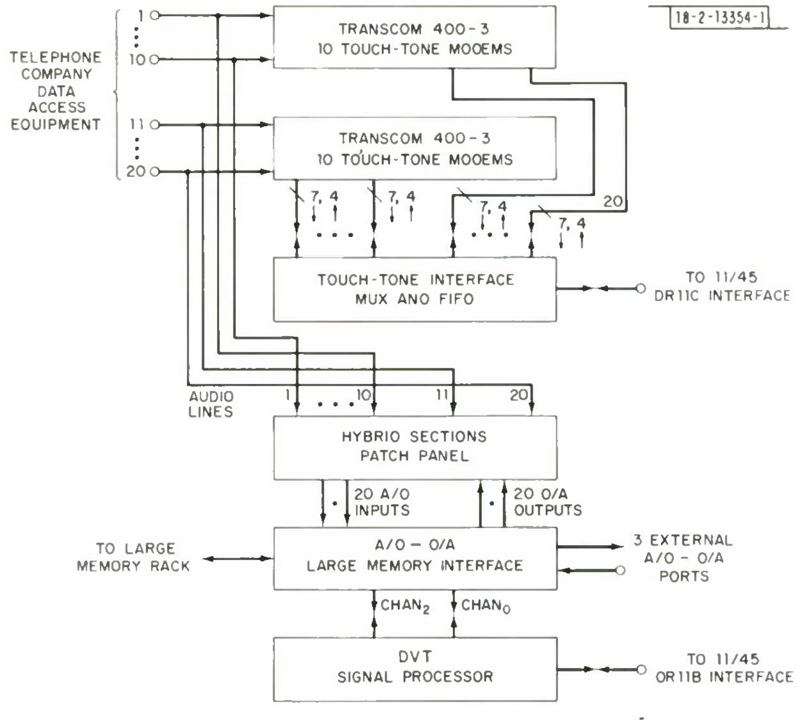

Fig. A-7. Conferencing rack.

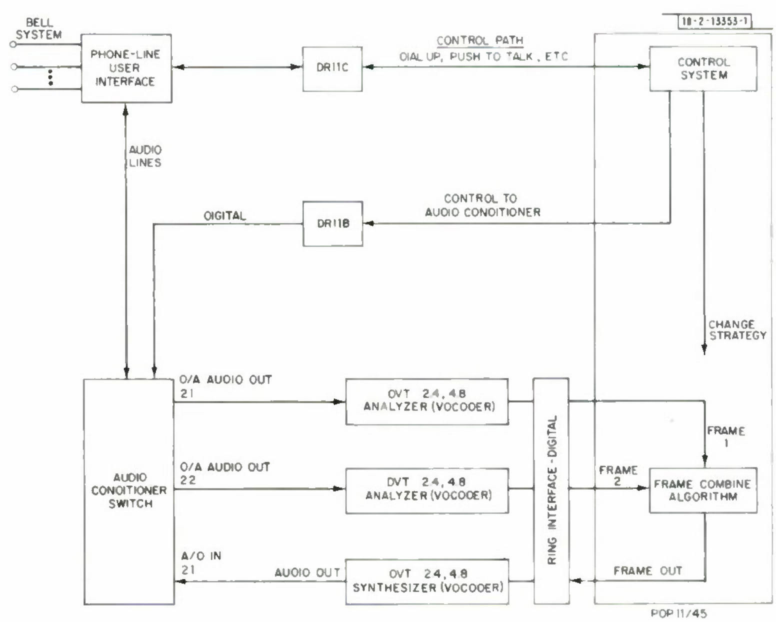

Fig. A-8. Frame bridging.

APPENDIX B
A RESOURCE ALLOCATION TASK

One of the primary tasks we expect to employ in evaluating teleconferencing systems is a complex scheduling task that we call "car pool."

In this task, each member of the teleconferencing group is given a map showing several cities and the driving times between them (see Fig. B-1). He or she is also assigned 1, 2, or 3 commuters, and is told in which city they live, where they work, and at what time they must arrive at work. The collective goal of the group is to form a set of car pools for the commuters that minimizes total driving time, with a penalty of 1/2 min. for each minute that a commuter arrives early for work. No commuter may arrive late for work, and individual car pools are limited to 3 commuters each.

Initial trials with this task showed that a great deal of the group's time was consumed in arithmetic computation rather than in interactive discussion of alternative solutions. This situation seemed undesirable, so a new approach was tried.

A computer program was written to generate possible schedules for various pairs and triplets of commuters. An example of such a schedule is shown in Fig. B-2. The first line of the schedule shows the optimal route and total score for commuter Brown traveling alone. Then a set of seven 2-commuter car pools is shown, where commuter Brown is the driver, along with the optimal routing and total score that would result in each case. Possible pairs in which the total score for the 2-commuter car pool is worse than the sum of the scores for the individual commuters driving alone are omitted. Finally, a set of nineteen 3-commuter carpools with Brown driving are shown. Again, possible triplets for which the score is worse than the sum of the individual scores are omitted.

Trials using these schedule sheets have proved far more satisfactory. A great deal of communication is necessary to formulate possible solutions. For example, if a 3-commuter car pool involving Brown, Cook, and Downs is proposed, the first step must be to determine which commuter should serve as a driver. To do this, the various permutations of the 3 commuters (e.g., BCD, CBD, and DBC) must be located on separate sheets by different members of the group, and the lowest score noted. Then, alternative combinations (e.g., Brown and Cook driving together, with Downs commuting separately) must be checked.

The same computer program that generates the schedule sheets also computes the overall optimal schedule, against which to compare the group's best solution. The program also counts the number of possible legal solutions that can be generated.

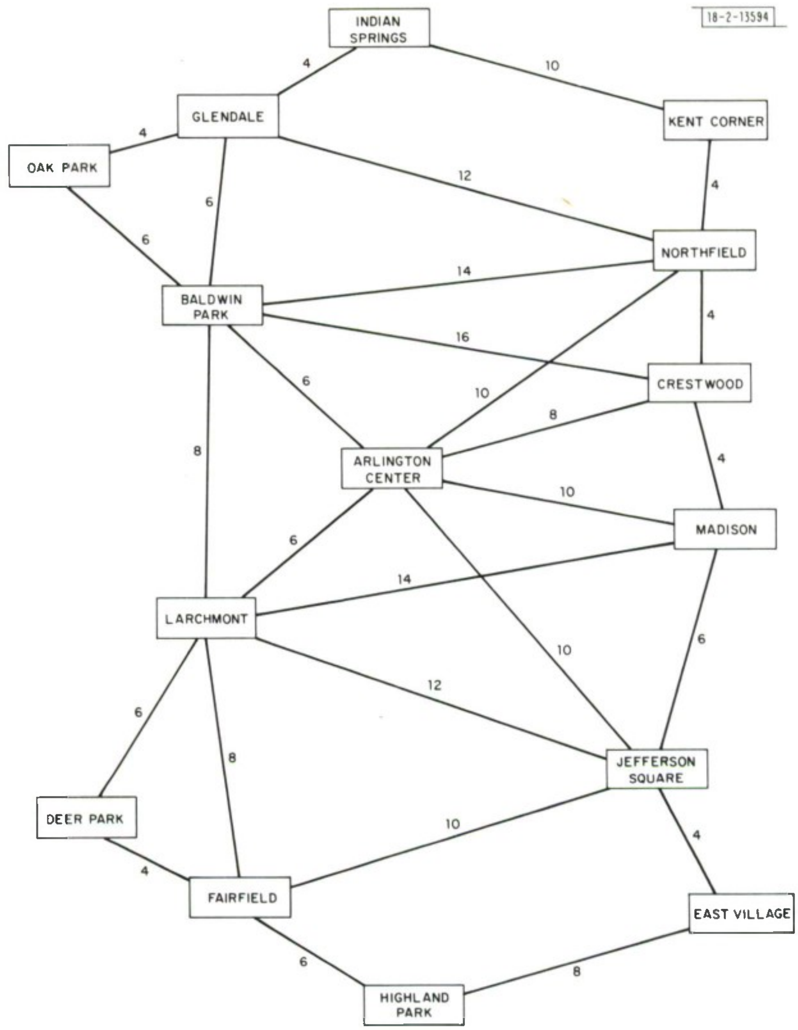

Fig. B-1. Map used in "car pool" task.

BROWN FROM FAIRFIELD TO NORTHFIELD BY 81Ø

COMS	ROUTE	DT	WT	TOT
B	FLAN	24	Ø	24
BC	FJMCNG	36	6	42
BD	FHEJMCN	32	5	37
BE	FDLACN	28	2	3Ø
BF	FHEJMCNKI	46	3	49
BG	FHEJMCN	32	9	41
BH	FLANKI	38	2	4Ø
BI	FJMCNK	28	2	3Ø
BCD	FHEJMCNG	44	17	61
BCE	FDFJMCNG	44	14	58
BCF	FHEJMCNGI	48	14	62
BCH	FLJMCNGI	5Ø	9	59
BCI	FJMCNKIG	42	16	58
BDE	FDFHEJMCN	4Ø	7	47
BDF	FHEJMCNKI	46	8	54
BDG	FHEJMCN	32	14	46
BDH	FHFLANKI	5Ø	9	59
BDI	FHEJMCNK	36	9	45
BEF	FDFHEJMCNK	54	5	59
BEG	FDFHEJMCN	4Ø	11	51
BEH	FDLACNKI	42	6	48
BEI	FDFJMCNK	36	6	42
BFG	FHEJMCNKI	46	12	58
BFH	FLJEJMCNKI	56	7	63
BFI	FJEJMCNKI	46	7	53
BGI	FHEJMCNK	36	13	49
BHI	FLJMCNKI	48	2	5Ø

-2-13595

Fig. B-2. Example of schedule sheet used in "car pool" task.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-77-162	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Network Speech Processing Program		5. TYPE OF REPORT & PERIOD COVERED Semiannual Technical Summary 1 October 1976 – 31 March 1977
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Melvin A. Herlin and Bernard Gold		8. CONTRACT OR GRANT NUMBER(s) F19628-76-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element No. 33126K
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Agency 8th Street & So. Courthouse Road Arlington, VA 22204		12. REPORT DATE 31 March 1977
		13. NUMBER OF PAGES 52
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Electronic Systems Division Hanscom AFB Bedford, MA 01731		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

network speech processing
secure voice conferencing
human factors methodology
packet speech
speech algorithms
bandwidth efficient communications
packet virtual circuits

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This is the first Semiannual Technical Summary Report on the Network Speech Processing Program to be submitted to the Defense Communications Agency. It covers the period 1 October 1976 through 31 March 1977 and reports on the following topics: Secure Voice Conferencing, Speech Algorithms, and Bandwidth Efficient Communications. Each of these tasks is directed to particular problems associated with AUTOSEVOCOM II and/or the design of future defense communications systems.